University of Portsmouth
Faculty of Technology
School of Engineering,

Artificial Immune System For The Detection of Abnormal Activity In Ambient Assisted Living

By

SEBASTIAN DOMINIK BERSCH
Dipl.-Ing., MEng

The thesis is submitted to the academic registry in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy of the University of Portsmouth

September 2014

Abstract

This thesis is concerned with the use of Artificial Immune System (AIS) in the area of Ambient Assisted Living (AAL). The hypothesis for the work presented herein is that the AIS features of self-learning and adaptability address the complex problem of improving the detection of unknown abnormal behaviour in the long-term monitoring of elderly.

The work presents an affordable Open Hardware Data Acquisition Device that in combination with a Markov chain-based software simulation environment can be used for the collection of human activity data and the generation of necessary information for long-term simulation. The main contributions from the work presented herein relate to the design and use of AIS based solutions, and the selection of appropriate parameter combinations for supervised classifiers. Firstly, a novel seeding technique for AIS is presented that improves the placement of detectors in the search space. Secondly, a novel AIS-based monitoring algorithm, inspired by Hierarchical Temporal Memory architecture, is designed to learn and approximate sensor data to detect and report activity abnormalities. Thirdly, an empirical analysis is carried out to provide a clear understanding of how sampling frequency, segmentation method, window size, and classifier method influence the Activity of Daily Living event classification accuracy and computational load in the area of AAL. Fourthly, a Pareto curve based technique has been devised and demonstrated as a useful tool for the informed selection of parameter combinations to achieve the best possible classification accuracy and computational load.

The evaluation of the AIS-based algorithm showed that the detection rate of abnormal activity outperformed the results of supervised classifiers with parameter combinations selected based on the Pareto curve. The results are encouraging and support the decision to introduce the use of AIS for the detection of abnormal activity in AAL environments.

Declaration

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are my own (the named candidate) and have not been submitted for any other academic award.

Acknowledgement

I would like to express my gratitude to the many people that helped me in one way or another during my time as a PhD student. First of all, I would like to thank my supervisors Dr. Djamel Azzi and Dr. Rinat Khusainov for their continued support and advice. Their feedback has been crucial in forming this work.

I would like to thank the staff in the School of Engineering who were always open to discussions and provided new ideas and inspiration. I am especially thankful to Mohamed Al-mosawi and Ifeyinwa Achumba who kept me company during long hours in the office.

I wish to thank all my friends for the support and the distractions they provided to make my time in Portsmouth so enjoyable.

Lastly I would like to thank my family for their continued support, and unwavering belief in me.

Table of Content

List of Figures

List of Tables

Dissemination

Journal Publications:

- S. D. Bersch, D. Azzi, R. Khusainov, I. Achumba, and J. Ries, "Sensor Data Acquisition and Processing Parameters for Human Activity Classification," Sensors, vol. 14, no. 3, pp. 4239–4270, Mar. 2014.
- S. D. Bersch, D. Azzi, and R. Khusainov, "Artificial Immune Systems for Anomaly Detection in Ambient Assisted Living Applications," International Journal of Ambient Computing and Intelligence, vol. 5, no. 3, pp. 1–15, 2013.

Conference Proceedings:

- S. D. Bersch, D. Azzi, and R. Khusainov, "Artificial Immune System - A New Approach For The Long-Term Data Monitoring in Ambient Assisted Living", 5th International Conference on Sensor Systems and Software (S-Cube), 2014. – In Press
- S. D. Bersch, D. Azzi, R. Khusainov, and I. E. Achumba "A Open Hardware Wireless Sensor Monitoring System for Human Wellbeing Research in Ambient Assisted Living", 5th International Conference on Sensor Systems and Software (S-Cube), 2014. – In Press
- S. D. Bersch, C. M. Chislett, D. Azzi, R. Khusainov, and J. S. Briggs, "Activity detection using frequency analysis and off-the-shelf devices: fall detection from accelerometer data," Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on, pp. 362–365, 2011.
- S. D. Bersch, D. Azzi, and R. Khusainov, "Fall Detection using Biologically Inspired Monitoring - Artificial Immune System in the Area of Assisted Living.," ECTA-FCTA, pp. 320–323, 2011.

Glossary of Abbreviations

Abbreviation	Meaning
AAL	Ambient Assisted Living
ABS	Anomaly Based detection System
ABSS	Agent-Based Social Simulations
ADA	ADaptive Agent
ADC	Analogue-to-Digital Converter
ADL	Activity of Daily Living
ADM	Activity Discovery Method
AI	Artificial Intelligence
AIRS	Artificial Immune Recognition Systems
AIS	Artificial Immune System
AmI	Ambient Intelligence
AmIE	Ambient Intelligent Environments
AMSL	Above Mean Sea Level
ANA	ANalysis Agent
ANN	Artificial Neural Network
ANOVA	ANalysis Of VAriance
APC	Antigen Presenting Cell
AR	Activity Recognizer
ARTIS	ARTificial Immune System
ASCII	American Standard Code Information Interchange
BT	Bluetooth
BUp	Bottom-Up
CA	Classification Accuracy
CAR	Circadian Activity Rhythm
CASAS	Centre for Advanced Studies in Adaptive Systems
CI	Computational Immunology
CL	Computational Load
CLONALG	CLOnal selection ALGorithm
CM	Classifier Method
CPS	Centre Point Seeding
CSV	Comma Separated Value
DAD	Data Acquisition Device
DCP	Detector Centre Point
Detector set R	Array containing matured Artificial Immune System detector
Detector set $R_{Immature}$	Array containing immature Artificial Immune System detector
Df	Degree of freedom (SPSS)
DIO	Digital Input Output
DM	Data Mining
DoF	Degree Of Freedom
DT	Danger Theory
DVSM	Discontinuous Varied-Order Sequential Miner
DWin	Dynamic Windowing
ECG	ElectroCardioGram
EEG	ElectroEncephaloGraphy
EGR	Exhaust Gas Recirculation
FAT	File Allocation Table

Abbreviation	Meaning
FFT	Fast-Fourier Transformation
FN	FalseNegative
FNSW	Fixed-size Non-overlapping Sliding Window
FOSW	Fixed-size Overlapping Sliding Window
FP	FalsePositive
FPGA	Field-Programmable Gate Array
FSM	Finite State Machine
FSRP	Fixed-Size Random Placement
GPS	Global Positioning System
HAR	Human Activity Recognition
HIS	Habitat Intelligent pour la Santé
HIS	Human Immune System
HMM	Hidden Markov Model
HTM	Hierarchical Temporal Memory
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
ISTAG	Information Society Technologies Advisory Group
KNN	K-Nearest Neighbour
LISYS	Lightweight Intrusion detection SYStem
LR	Logistic Regression
MAS	Multi Agent System
MCC	MultiClassClassifier
MEMS	Micro-EletroMechanical System
MISO	Master Input Slave Output
MIT	Michigan Institute of Technology
ML	Machine Learning
MOSI	Master Output Slave Input
NS	Negative Selection
OCSVM	One Class Support Vector Machine
OHDAD	Open Hardware Data Acquisition Device
PCB	Printed Circuit Board
PDA	Personal Digital Assistant
PDF	Probability Density Function
PIR	Passive InfraRed
RbW	Reference-based Windowing
RFC	Request For Comment
RFID	Radio-Frequency IDentificaiton
RMS	Root Mean Square
RTC	Real-Time Clock
RTLS	Real-Time Location System
Rx	Receive
SAX	Symbolic Aggregate approXimation
SCL	Serial CLock for I2C
SCLK	Serial CLocK for SPI
SDA	Serial DAta
SEA	SEnse detection Agent
Self-element	An element out of the self-set S or S^*
Self-expert	Having enough knowledge of a common field to define system behaviour

Abbreviation	Meaning
Self-non-self	The process to differentiate between normal and abnormal data.
Self-set S	Data Set used to train the Artificial Immune System. Should only contain normal data
Self-set S^*	Data Set used to test the Artificial Immune System. Contains normal and abnormal data
SF	Sampling Frequency
SHA	Self-Healing Agent
Sig	Significance (SPSS)
SM	String Matching
SM	Segmentation Method
SMA	Signal Magnitude Area
SMD	Surface Mounted Device
SMO	Sequential Minimal Optimization
SPI	Serial Peripheral Interface
SS	Serial Select
STD	STandard Deviation
SVM	Signal Vector Magnitude
SVM	Support Vector Machine
SWAB	Sliding Window And Bottom up
THT	Throughput Hole Technology
TI	Texas Instruments
TN	TrueNegative
ToD	Top-Down
TP	TruePositive
Tx	Transmit
UART	Universal Asynchronous Receiver / Transmitter
UCI MLR	University of California Irvine Machine Learning Repository
UMASS	University of Massachusetts
UWB	Ultra Wide Band
VDE	Verband Der Elektrotechnik, Elektronik, Informationstechnik e.V (Association for Electrical, Electronic & Information Technologies)
VSRP	Variable-Size Random Placement
VSW	Variable-size Sliding Window
WS	Window Size
WSN	Wireless Sensor Networks

1 Introduction

1.1 Foreword

This chapter provides the necessary context that defines the research confinement of the work carried out in this thesis. It discusses the demographic change and expected consequences that have been identified by western governments and which forms the application problem addressed by the work presented herein. The chapter also provides a brief background to the concept of Ambient Assisted Living (AAL) and Ambient Intelligence (AmI). Thereafter, the rationale and motivation behind the presented work are discussed in detail. The chapter concludes with a statement of the objectives of the research work underlying this thesis and an overview of the latter.

1.2 Background

Over the past decades, the advances in healthcare and technology have steadily increased the prospect of a longer healthy life for western citizens. The ensuing demographic change has not only had an impact on humans but also on the society and the economics of the countries they live in. In 2009, the EU commission estimated that in 2060 half of the population will be above the age of 48. This includes 151 million people above the age of 65 (from 85 million in 2008) and 61 million above the age of 80 (from 22 million in 2008).

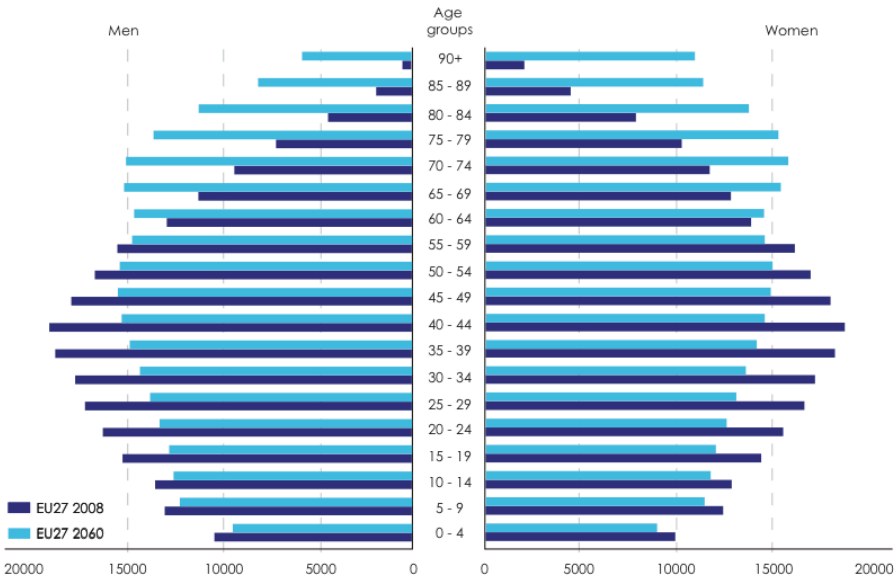


Figure 1–1: Age population pyramids (in thousands) from the EU 27 in 2008 and 2060 (Source: [1])

Figure 1–1 illustrates the predicted shift between the age pyramids of 2008 and 2060. The increase in elderly combined with a decline in fertility and available labour supply will result in higher spending required from the government to support public healthcare [1]. A reflecting report published three years after the initial investigation, showed that the actual age-related expenditures in 2010 was in fact higher than expected, showcasing the seriousness of the demographic change. Moreover, the report highlights that public spending on long-term care will most likely double by 2060 [2]. Similar numbers are predicted by the WHO, who estimate 2 billion people to be over the age of 60 by 2050, compared to the 605 million in 2000 [3]. In earlier decades, elderly people would normally be looked after in a private household. The children or in-laws, who would usually live close by or even share the same house in rural areas, would provide all or part of the required care. Through industrialization and the following acceptance of women in the work force, an important foundation of the traditional eldercare broke away. This was aggravated by the required mobility of the worker, which resulted in greater geographical distances between families (including elderly and their offspring) [4]. It is therefore a necessity to find alternative options for the eldercare, which has the aim to supplement the traditional care, allow for longer independent living at-home and reduce governmental costs for the increased aging population. There exist a vast amount of possible options to support and improve the healthcare system. In [5] the author separated the area in six different aspects with their corresponding responsibilities as represented in Figure 1–2.

	Emergency Treatment Services	Autonomy Enhancement Services	Comfort Services
Indoor Assistance	Emergency Prediction Emergency Detection Emergency Prevention	Cooking Assistance Eating Assistance Drinking Assistance Cleaning Assistance Dressing Assistance Medication Assistance	Logistic Services Services For Finding Things Infotainment Services
Outdoor Assistance	Emergency Prediction Emergency Detection Emergency Prevention	Shopping Assistance Travel Assistance Banking Assistance	Transportation Services Orientation Services

Figure 1–2: Aspects of healthcare as presented by [5]

The work described in this thesis limits its scope to the Emergency Treatment Services, with the focus on indoor assistance. [6] reports a trend that is leading away from long and expensive hospital stays to an earlier discharge with the support of a personal

response system for private homes. This is reinforced in the analysis of [7] who found out that most elderly would prefer to stay in their own home, even with the help of technology. Moreover, such a care system can support care personnel and supplement personal care in the home and postpone the need for an expensive nursing home that would offer 24 hour monitoring services. Additionally, a technological support system allows for peace of mind, not only for the elderly but also for the family members that are not in close proximity to actively participate in the care process. This in-home monitoring allows for a vast amount of research and business opportunities and hence several commissions and consortia have been established and funded to look into this area to propose design standards, highlight possible research areas and fund possible solutions under the so called Ambient Assisted Living (AAL) umbrella. The German Association For Electrical, Electronic & Information Technologies (VDE) defines AAL as: "... a hybrid product referring to a basic technical infrastructure in the home and services provided by third parties with the aim of continuing to lead an independent life in one's own home."[8]. The in-home approach does not need to be restricted to one's own home only, as Internet technologies allow for information distribution [9]; other context-aware Ambient Intelligent Environments (AmIE) could adapt to visitors based on shared privat information.

1.3 Ambient Assisted Living

The European Commission proposed the Ambient Assisted Living (AAL) initiative as part of the Ambient Intelligence (AmI) concept established by the Information Society Technologies Advisory Group (ISTAG) of the European Commission [10]. In [11] the authors speak of an invisible intelligent environment that senses, adapts, and reacts seamlessly to users in their normal environment (house). Another definition for AAL comes from [12]: "AAL refers to intelligent systems of assistance for a better, healthier and safer life in the preferred living environment". These definitions in combination with the definition from the VDE highlight, that AAL is not only for the elderly but tries to improve the living standard in general for the home or preferred environment. That AAL is not only an academic research area, is highlighted by the market analysis report from the Frauenhofer Institute, which shows high potential for the world market [7]. Especially when the use of AAL reduces the cost of hospital stays and might even lead to voluntary acceptance of family members instead of expensive care homes [13].

The requirement to combine the flawless interaction of several core-technologies with human wellbeing, which is a complex and individual mental construct [7], makes AAL a complex research field. The German VDE [8] tries to break down this complex area into four main areas: Communication, Mobility, Self-sufficiency, and Life at home. Furthermore, in [14], the authors suggest that AAL should provide health and comfort oriented services. The target group for AAL services are people in the age of 55+, as they are prone to chronic illnesses [7]. Even though, a sensor equipped home environment is designed with the intention to support the elderly, it is also sometimes seen as a golden cage. [6], [7] report that the feeling of being constantly monitored can sometimes result in negativity against the system. Contrary to the belief that AAL is designed to replace personal interaction with health care personnel, it is more a tool to support and alarm in emergency situations. The Philips Research lab [15] points out that each technological achievement has to overcome a natural reluctance of people to adaptation until the real benefit of the new technological step are understood and trust and comfort can be build. AAL can start changing lives from this point onwards.

1.4 Ambient Intelligence

AAL can be seen as a top-layer sitting on top of Ambient Intelligence (AmI) with the addition of specific healthcare knowledge and the aim to support the elderly. The underlying AmI methodology needs to be understood, to allow for a better understanding of AAL. Figure 1–3 illustrates a simplified block diagram of the software paradigms involved for AAL. [16] describes the goal of AmI to be unobtrusively hidden in everyday lives whilst adjusting seamlessly to a person's individual need. Philips Research Lab [17] estimates that with the current developments of ubiquitous / pervasive computing, AmI will be a reality in ten to fifteen years. What this highlights, is that AmI is actually a combination of several different computer paradigms. The two 'basic' elements that provide the hidden computer processing power and data input/output interfaces are Ubiquitous and Pervasive Computing. Both terms are usually used interchangeable, while sometimes pervasive computing is more geared towards the user-to-machine interaction and ubiquitous computing focuses on the hardware side.

The sensors and microcontrollers for the processing power can be found in everyday equipment such as toasters, washing machines, smart meters, smart garments, etc. These two architectures allow for the collection of environmental data and the interaction with

the environment. To make sense of the vast amount of collected data and provide context-aware interactions, some sort of intelligence is required for AmI. Software agents are therefore based upon Artificial Intelligence (AI), Machine Learning (ML), or Data Mining (DM). This thesis is mostly focused on the AI, ML and DM side of AAL and is therefore not focusing on Ubiquitous and Pervasive Computing.

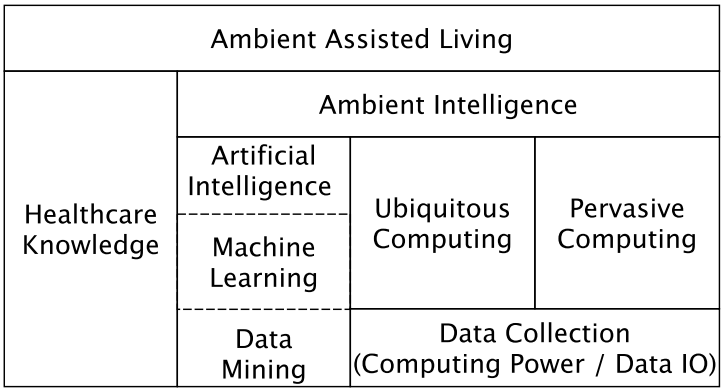


Figure 1–3: Software paradigms involved in Ambient Assisted Living

The beginning of AI can be traced back to the early days of Greek philosophers such as Aristotle with his research in philosophy and his belief in logic, building the foundation for todays modern AI comprehension. With the beginning of the computer era around 1940 and Alan Turing's contribution of the Turing Test in his research paper in 1950 [18] the idea of thinking machines was introduced to a wider field of researchers. Six years later, at a workshop at the Dartmouth College in 1956, a group of 10 researchers introduced the term Artificial Intelligence, as it is perceived today. As described in [19], AI is built upon the fields of Philosophy, Mathematics, Economics, Neurosciences, Psychology, Computer Engineering, Control Theory, Cybernetics, and Linguistics. Modern AI research moved away from isolated individual research areas and is now moving to combine different research fields, to deal with uncertainty in data. While AI aims to produce intelligent agents that can learn, adapt, and react to an uncertain environment, ML and DM are more concerned with the underlying data model. In [20] DM is described to be the core component of knowledge discovery in large datasets. DM uses algorithms to infer a data model, which helps to discover and understand unknown patterns in the data. From this knowledge future predictions regarding the data can be drawn. While DM is used to discover patterns in the data, ML is used to predict future data. ML is normally presented with either labelled test data (supervised) or unlabelled (unsupervised) to build a data model and predict unlabelled future data. ML is different from DM in the

sense that it does not try to discover patterns but learns from the available data. As all three fields build upon each other, it is hard to draw clear lines around the subject areas.

1.5 Rationale

One of the highlighted problems for AAL in the AALIANCE Roadmap [12] is intelligent reasoning. Without reasoning a monitoring system cannot make any intelligent prediction about the health state and wellbeing of an elderly. Considering a use-case such as activity monitoring for health, accident, and fall prevention presented by [7] and [12], reasoning is most important. Especially since [21] highlights the increased risks of elderly to fall, and as a consequence thereof, the increased possibility for injuries or even death if a fall stays unnoticed. So what does such a use-case actually entail?

The scenario of abnormality detection relies on electronic sensors distributed in an environment. These sensors can be on-body worn or off-body sensor devices, such as accelerometer, magnetometer, ECG, PIR, noise, temperature, either individually or as a combination. This sensor data represents the behaviour of the person. If this information points to either a known problem or indicates a deviation from the norm, an alarm can be raised to contact emergency personnel to allow for a faster response time and therefore faster live saving measures [7]. A dynamic model of learned behaviour should be preferred, as human beings get older, their behaviour changes over time, such as longer sleep, increased usage of the toilet, or age related gait changes. It is therefore required to update a learned model and adapt to changes over time [7], [12]. Research has shown, that there are different methods and techniques to try and make sense of collected data, mostly concentrating on supervised learning.

Additionally, there exists a lack of openness in the research field. [7] mentioned missing open standards and an open interoperability between systems. Different AAL systems might not work with each other and lead to counter productivity. The authors rally for an “Open-Source” AAL-reference platform to allow for a speedier and unified research results. It is further mentioned that research results should also be openly shared under a similar open-source license.

1.6 Motivation and Objectives of the Research Work

The motivating factors for the work presented in this thesis are the emerging challenges in the new area of Ambient Assisted Living (AAL), with respect to:

- Problem of open-hardware for data acquisition

- Unavailability of long-term data
- Simplified long-term data analysis
- Optimal parameter selection for classifiers

Long-term data analysis is one of the most important challenges in the future of AAL, to enable elderly to live longer independently, in an accustomed environment. It is therefore the aim of the work reported in this thesis to introduce the use of Artificial Immune System (AIS) for the long-term monitoring, to identify abnormalities in the elderly movement to detect falls without prior training.

The aim is to introduce a new paradigm to the filed of activity monitoring for AAL through the use of AIS. The new paradigm has the advantage that it does not require a dedicated training phase but is constantly learning and evolving, allowing adaptation to the behavioural changes of an elderly person. The usability of AIS is proven as part of this work and compared to supervised classifiers used so far in AAL. The specific objectives of the work reported in this thesis are:

- Provide an affordable data acquisition platform for the collection of human activity. All information required for the reproduction should be made available to other researches to allow adaptation and modification to fit other needs.
- Reduce the time required to generate different long-term behaviour scenarios using the above data acquisition device in combination with a Markov chain-based simulation environment
- Compare the detector seeding methods used for Artificial Immune Systems (AIS) and develop a novel seeding approach based on the time of arrival of sensor data.
- Develop an AIS-based monitoring algorithm for the detection of abnormal activity using accelerometer sensor data collected by the data acquisition platform.
- Contribute to a better understanding of the influence of sampling frequency, segmentation method, window size, and classification method on the Activity of Daily Living (ADL) event classification accuracy and computational load in the area of AAL. The analysis should provide a tool for the informed parameter selection for supervised classifiers already used in the ADL event classification.
- Proof the usefulness and importance of AIS in AAL to present it as a novel method for the detection of abnormal activity without prior training. This part will also compare the computational load required by the different methods to analyse a

single data segment in order to verify the use for the real-time sensor data analysis.

The combination of these objectives forms the basis of the implementation of the research work.

1.7 Overview of the Thesis

This chapter has described the background, motivation and objectives of the work presented in this thesis. The rest of the thesis is organized as follows:

Chapter 2 presents the review of literature for the data generation in AAL environments. The chapter also details the different short- and long-term monitoring approaches in the field before introducing the use of Artificial Immune Systems and its use in abnormality detection in other areas. The chapter is concluded with a summary.

Chapter 3 describes the Open Hardware Data Acquisition Device (OHDAD) developed as part of this work, the implemented software simulation environment based on Markov chains and the evaluation process and results.

Chapter 4 provides an introduction to the Artificial Immune System (AIS). The different aspects of the system are presented and different detector seeding methods are explained and compared. Afterwards the monitoring algorithm based on the AIS, implemented as part of the work presented here, is described.

Chapter 5 details the empirical investigation carried out to provide a better understanding in the parameter selection for supervised classifiers, as the literature review showed significant variation for the sampling frequency, segmentation method, and window size used.

Chapter 6 presents the comparison between the monitoring algorithm based on AIS and other supervised classifiers. Parameters for the supervised classifiers are based on the results obtained in Chapter 5.

Chapter 7 summarises the thesis and discusses the limitations, and contributions of the work. The chapter also presents suggestions for future work and a conclusion.

1.8 Summary

The background, motivation, and objectives of the work presented in this thesis have been given, followed by an overview of subsequent chapters. The next chapter presents a review of literature on the various aspects of the work presented here.

2 Literature Review

2.1 Introduction

This chapter presents a summary of previously published work which forms the background for the work presented here; namely, the introduction of Artificial Immune System for the detection of abnormal activity in Ambient Assisted Living. More specifically the literature review covers the following areas:

- **Data Generation** – presenting existing data sets, sensors and hardware for the data acquisition, simulation environments, and background in data processing.
- **Monitoring** – introducing the different approaches to short- and long-term data monitoring.
- **Artificial Immune Systems (AIS)** – describing AIS, and its various fields of application.

2.2 Data Generation

For the design and evaluation of any monitoring system, an understanding of the available sensor data and its representation is required. A survey of the literature revealed different approaches to get to this vital information. Data acquisition is normally executed under laboratory conditions or in house-like environments equipped with sensors (test beds and smart houses), the latter either belong to the University or are the participants' own homes. As there exists no mathematical model for the complex system of a human, no complete simulation environments exist. The following sections discuss the sensor data sets available in the public domain, the sensor platforms used for the data acquisition, as well as existing simulation environments.

2.2.1 Open Data Sets

When looking into the comparison of Machine Learning (ML) or Data Mining (DM) related algorithms, performance measures are normally calculated using benchmark data sets, such as the Iris data set found in the University of California Irvine Machine Learning Repository (UCI MLR) [22]. To the author's best knowledge, no such data set exists for AAL. Additionally, there is no database that hosts exclusively AAL related data sets from different research groups, which would allow sharing, redistributing, comparing, and validating research outcomes. Only recently, the authors of the

Opportunity data set [23] attempted to introduce the idea of a benchmark data set. It contains 10 different types of sensors worn on- and off-body and totals to 72 sensors. Figure 2–1.a illustrates the on-body sensor positions and their sensor type. Unfortunately, the authors only concentrated on normal activities. In total, data from 12 test subjects were recorded but only three test subjects have been annotated to date. The Northeastern University [24] collected a similar data set. The data set contains the acceleration data that was measured simultaneously at five different body positions, illustrated in Figure 2–1.b, of 20 test subjects [25]. As before, this data set only concentrates on normal activities. Figure 2–1.a and Figure 2–1.b illustrate the various on-body sensor positions highlighted by literature. The question that is not answered in the relevant research papers is the best position for the data collection. The different positions have to be considered during the design stage of the Open Hardware Data Acquisition Device (OHDAD) in Chapter 3 and are here illustrated for a better visualisation for the reader. A third data set is provided by [26], the authors collected XY position coordinates from a person moving in a laboratory environment, using the commercially available Real-Time Localization System (RTLS) from Ubisense.

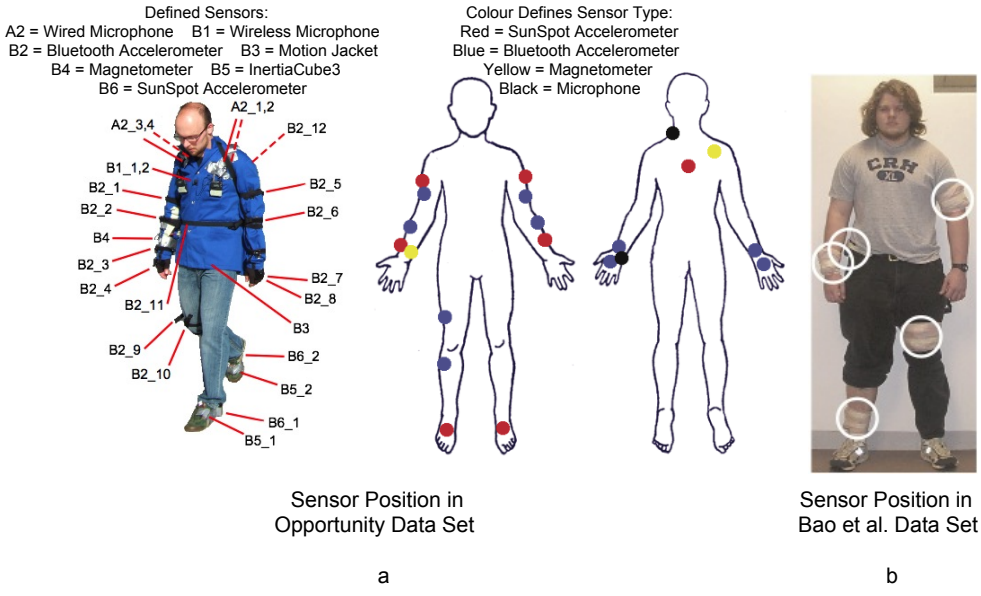


Figure 2–1: On-body sensor positions for the Opportunity and Bao et al. data set (Source: [23], [25], [27])

2.2.2 Sensors and Data Acquisition Devices

In the Ambient Assisted Living Roadmap [12], the key technologies and functions for AAL are defined as: sensing, reasoning, acting, interacting through interfaces, and

communication. For the data generation, only sensing is of interest. The roadmap continues to state that sensing should "take place in anything and anywhere: in- or on-body, in- or on-appliances or in the environment". As this list is quite extensive, the literature search was limited to on-body, and environment sensors for indoor use.

Table 2–1: Types of sensors used in the area of Ambient Assisted Living

Body Worn Devices	Environment Devices
Accelerometer	RFID
Gyroscope	Infrared (PIR)
Mercury Trigger	Ubisense
Inclinometer / tilt Sensor / Switch	Pressure Mat
GPS / Compass	Accelerometer
Sole Pressure of Foot Switch	Pressure Sensor
RFID	GPS
Ubisense RTLS	Microphone
Infrared (PIR)	Contact Switch
ECG / Heart Rate Monitor	
EMG	
EOG	
Skin Conductivity	
Body Temperature	
Magnetometer	
Barometer	
Oximeter	
Micrphone	
Humidity Sensor	

In [28] the authors presented an extensive review of over 160 research papers and the corresponding sensors that were used (presented in Table 2–1). Their conclusion is that for fall detection the most body worn sensors are accelerometers, gyroscopes, and tilt sensors while for environment placed devices a standard selection consists of Passive InfraRed (PIR) sensors, microphones, and pressure mattes. In the context of this work, sensors are only one component for Data Acquisition Devices (DAD) and some of the more general-purpose DADs for AAL should be highlighted.

The DAD presented in [29] uses two bi-axial accelerometers to emulate a tri-axial accelerometer. The authors only provide a block diagram of the system indicating a wireless network module but no schematics regarding the microcontroller or any information about the software running on the microcontroller. Another system based on a bi-axial accelerometer, is presented in [30]. Their work uses an on-body DAD for the activity and location recognition. The system includes a Personal Digital Assistant (PDA) as an interface and a microcontroller for the data collection. Additional sensor data is collected from a digital compass sensor and a gyroscope. The authors only provide a picture of their complete setup, without any further information of the wiring, block diagram or software workflow.

In [31], the authors stated that a vast amount of literature relies on accelerometer data in the area of AAL. They continued to criticise that most of the reviewed research papers use data collected under laboratory conditions, which might not correspond to real life scenarios. Their system collects pictures from a USB camera and communicates with a Bluetooth (BT) accelerometer as an additional peripheral. The authors do not explain if the BT connection can be used for additional data transfer or if it is only for the communication with the accelerometer. The system can be used for up to 4 hours with a portable lithium battery. It is worn by the user on the torso and requires no further interaction after turning the device on.

In [32] the authors highlighted that Wireless Sensor Networks (WSNs) will boost the use of wearable medical devices aimed to advance the general wellbeing and fitness of every person. The authors approach to remotely monitor the ElectroCardioGram (ECG), blood pressure, and pulse rate values of an individual is based on the Tyndall25 platform [33]. Wireless communication is provided by an ISM band 2.4 GHz transceivers. The main processing is executed on a low power 8-Bit microcontroller with 128 kB of program memory. The authors do not provide any further information about the hardware platform, besides some diagrams of the peripheral amplification circuit. In addition to this, their selection of ISM transceivers requires special hardware or another base station to let it communicate with modern laptops or smartphones.

The available data set mentioned in Section 2.2.1, provided by [25] was collected using Hoarder boards. These boards were designed at the Michigan Institute of Technology (MIT) for sensing and manipulating environments with extremity-computing devices [34]. The board collects bi-axial accelerometer data to monitor users whilst they carry out different Activities of Daily Living (ADLs). On the project website [35], the author provided all necessary files to reproduce a Hoarder board but stated himself, that the board is out-dated by today's standards. It relies on CompactFlash cards, which have since been superseded by smaller SD and Micro SD Cards, the serial port for the data communication is not available on modern laptops anymore, and the need for a Windows machine limits researchers to use only specific equipment.

The authors of [36] presented a wearable wireless accelerometer device for fall detection. The authors pointed out that the miniaturization and cost efficiency of today's Micro-ElectroMechanical Systems (MEMS) sensors allow for new economical monitoring. The hardware is based on a Field-Programmable Gate Array (FPGA) Spartan

3, a tri-axial MEMS accelerometer, and a ZigBee transceiver for wireless data transmission. The FPGA is programmed using a USB connection, available on all modern PCs. The accelerometer is sampled at 10 Hz and has a range of ±2 g. Wireless communications transmits each axis acceleration value as a string using a Universal Asynchronous Receiver/Transmitter (UART) controller. A base station time-stamps received data and controls the data fusion from additional sensors.

The data set generated by [26] and downloadable from the UCI MLR website, uses Ultra Wide Band (UWB) radio waves. The commercial hardware is distributed by the company Ubisense and consists of off-body sensors. These sensors allow the real-time localization of active on-body tags in a 3D environment. The position is estimated using the radio waves' time difference of arrival and the angle of arrival. The authors reported a tracking accuracy of about 15 cm over 95% of the samples, when limited to 4 tags and a 10 Hz sampling frequency.

2.2.3 Data Segmentation

2.2.3.1 Introduction

The literature covering the data acquisition devices leaves out a crucial aspect of the event classification for Ambient Intelligence Environments (AmIE), namely, information pertaining to data acquisition conditions. Before any sensor data can be used for the analysis to draw conclusions with regards to the environment, certain necessary pre-processing steps have to be carried out. The following section therefore presents current and past work regarding the data acquisition, segmentation and windowing schemes, which need to be adopted before further data processing can take place.

2.2.3.2 Sampling Rate

The acquisition of data is one of the most critical steps in event classification as re-running experiments with test subjects is not always possible. Undersampling leads to loss of information and oversampling can result in information buried in unwanted noise. In the latter case, longer computational time is needed for analysis as more data needs to be processed. The minimum sampling rate $f_{sampling}$ is dependent on the maximum frequency contained in the data signal f_{max}, as the sampling theorem $f_{sampling} \geq 2 * f_{max}$ has to be fulfilled.

The highest sampling rate for AAL that the author found during the literature review is 512 Hz by [37] followed by the works of [38], [39] where the authors use a sampling rate of 256 Hz to collect accelerometer data. [25] use a bi-axial accelerometer and a sampling frequency of 76.25 Hz, which is less than 1/3 of [38] sampling rate. [40] choose $f_{sampling}$to be 64 Hz. The authors acknowledge the high sampling rate used by [38] however they reduced the sampling frequency on the basis that lower values are more feasible with off-the-shelf activity monitors. They further mention the work of [41], who sampled accelerometer data at 50 Hz, therefore resampling their own data at the same frequency as well. Overall the literature highlights that values around 50 Hz are one of the more common sampling rates. [31] use 52 Hz, [42] use 50 Hz to sample their tri-axial accelerometer, while [43] and [44] also report a 50 Hz sampling rate for an eWatch with bi-axial accelerometer and a light sensor. To the author's best knowledge, the work in [44] is the only one that tested different sampling frequencies (from 1 Hz to 30 Hz) for the sensor data. The outcome highlights that the recognition of ADLs improves with higher sampling rates but only marginally improves with sampling rates above 20 Hz. In [45] the authors demonstrate that 98% of the FFT spectrum amplitude is contained below 10 Hz, and 99% below 15 Hz. This corresponds to the findings of [46] who state that a sampling frequency of 20 Hz is sufficient to successfully classify ADLs. The lowest sampling rate that was found in the literature is 5 Hz by [47].

2.2.3.3 Data Pre-Processing Techniques

2.2.3.3.1 Segmentation Method

One of the challenges of data pre-processing following acquisition consists in deciding which points to actually use in the live stream of data. Several different segmentation methods exist to divide a larger data stream into smaller fit for processing chunks. The selection of the right segmentation technique is crucial, as it immediately impacts on the extracted features used for the ADL classification and the resulting classification accuracy. Therefore even the best classifier's performance will be poor when the extracted features are non-differentiable [48]. Furthermore, the segmentation techniques can also have an impact on the real time capabilities as complex segmentation methods can increase computational load but might result in improved classification accuracy. Moreover, the segmentation method also dictates how often features need to be extracted and classification algorithms need to be executed.

Literature has highlighted several different segmentation techniques used in various research projects, such as: Fixed-size Non-overlapping Sliding Window (FNSW) [48], [49]; Fixed-size Overlapping Sliding Window (FOSW) [48], [49]; Top-Down (ToD) [49]; Bottom-Up (BUp) [49]; Sliding Window And Bottom-up (SWAB) [49]; Symbolic Aggregate approXimation (SAX) [48]; String Matching (SM) [48]; Reference-based Windowing (RbW) [50]; Dynamic Windowing (DWin) [51] and Variable-size Sliding Window (VSW) [52]. The significant difference in these techniques resides in their online and offline capabilities. The meaning of an online technique is that the data can be segmented before the complete data is collected, while offline methods require the entire data set first. For real time applications, only online techniques are of interest. [49] note that online algorithms can produce very poor approximations of data under certain conditions but have a relatively good performance on noisy data. However, the authors also highlight that the FOSW segmentation algorithm is of particular interest in medical research, e.g. patient monitoring as the algorithm is simple and intuitive for researchers to understand.

2.2.3.3.2 Window Size

Researchers who use fixed-size window segmentation methods apply inconsistent window sizes. In [31], the authors used especially short windows of 1 s. [40] report to have used a 2 s window based on their short ADLs in their research and because they achieve only a minimal gain in classification accuracies with features from a 3 s window. Further examples for short windows are [53] and [41], with 2 s and 2.56 s, respectively. [44] extract features from a 4 s buffer, [43] use 5 s in their research, and [25] report a window length of 6.7 s. While these researchers are using short window sizes, [52] describe the usage of a 60 s windows in the work of [54] and 74 s in [55]. Furthermore, [52] introduce possible modification to the fixed-size window methods. The authors suggest in their work to dynamically adjust the window size based on special events in the sensor data as different ADLs have different time frames. They raise the point that longer window sizes can cover more than one ADL, while a small window could split an activity, which both leads to suboptimal information for an activity classifications algorithm. The work of [47] has a similar point, indicating that to achieve good classification accuracy, different sensor features should be extracted using varying window sizes. These methods lead to complex monitoring systems if several ADLs need

to be classified. Each feature window could yield different ADL classification results, which would then require a voting system to predict the correct ADL from the list of possible activities.

2.2.4 Simulation Environments

The problem with the data collection using DADs is the high investments for AAL laboratory facilities, which are not possible for every research facility. The data collection also requires ethical consideration, especially when elderly test subjects are included in a test group. Additionally, the collection and labelling of data requires a long time and as it cannot be speed up, delays the further data analysis (also called lead time) [56]. These problems could be drastically reduced with the use of a simulation environment. Standardised simulations packages would enable research groups to better compare their algorithm performances. The simulation environments of interest for the work presented here should simulate Ambient Intelligent Environments (AmIE) and the included sensor and/or human test subjects.

In [57] the authors described a simulation environment that concentrates on smart homes. The software is designed to support the research of service development, sensor deployment, and personalization in AmIE. The main advantage the authors highlight is that the software allows them to add new devices into the system, without buying or physically developing it. It therefore saves them time and money. Furthermore, it allows for a more intense evaluation of designed services.

The University of Massachusetts (UMASS) developed an environment (called Multi Agent System (MAS)) using intelligent agents [58]. Each agent simulates a home appliance like, water heater, coffeemaker, heater etc., and a robot to physically manipulate the environment. Every agent consumes resources, such as hot water, electricity, and (produces) noise, during runtime. As resources are only available in a finite number and the noise level should be kept to a minimum, agents consuming the same resources are competing against each other. The resource usage is used as a performance metric to test and refine developed agent control algorithms and communication protocols for the AmIE. The simulator helped to build a real test bed, called IHome, at UMASS. With the real test bed and the simulation environment the differences between these two testing methods can be investigated.

In [56] the authors describe the challenges of AmIEs to be expensive in development, time consuming, and hard to find human test subjects. The authors state that without an AmIE, research is limited and achieving significant impact exceptionally complicated. The paper continued to state that a simulation of an AmIE could be a possible alternative. This lead to the development of Persim. It has the aim to simulate human activities in a smart environment based on sensor events. Through the simulation environment, different AmIE layouts can be designed and tested. The human activities are defined using an activity mapper that links the execution of an activity to a list of predetermined sensor activations (such as motion sensors). Persim supports an Activity-driven and State-space simulation mode. The former generates a data set based on the defined activities using the sensors that are linked to a specific activity, while the latter generates a data set using the sensors that are independent of an activity. The simulation outcome was verified using a fuzzy agent that compares the simulated data set with real collected data from a three-bedroom apartment that is part of the Centre for Advanced Studies in Adaptive Systems (CASAS) project at the Washington State University.

A different simulation direction is presented in [59]. The authors' research investigated the simulation of agents in virtual environments. When trying to understand one-to-many interactions such as in evacuation research, an agent's behaviour needs to be diverse in order to allow for realistic simulations. The behaviour models used are based on a participatory simulation in which the physical information (movement of the agent in the environment) is stored and the subject is interviewed afterwards to receive domain knowledge for the earlier observation. The combination of the observation and domain knowledge are then used to build an operational model of the subject.

Human behaviour is also required for the simulation of medical interventions. In [60] two models are presented that look into the screening for different diseases in combination with patient behaviour. The goal is to improve the simulation for the evaluation of medications, as unpleasant medical side effects can result in patients dropping out before the end of a study. In an earlier study in [61], the authors included the physical state, emotions, cognitions, and social status of the test subjects in their behaviour model. Their initial model was widely tested and successfully applied outside the healthcare sector. However, the authors reported that their work requires further empirical research to validate the model and variable parameters used.

The problem with the studies above is that they are not designed for the investigation of behaviour abnormalities in a wellbeing context. As part of the French 'AILISA' project [62], researchers at the Habitat Intelligent pour la Santé (HIS) smart home, investigated a human behaviour simulation for abnormal health conditions. The benefits would be to produce enough data that would allow the evaluation of algorithms without waiting month or even years for the collection and labelling of data. In [63], the authors highlight the low number of available literature on human behaviour. Furthermore, literature that was found to cover a person's behaviour was aimed at psychological research rather than based on a mathematical model. Therefore, they based their behaviour model on Markov chains and then simulated the deviation of the patients' physiological parameters. In [64], the probability values for the Hidden Markov Model (HMM) were derived from real data. During a month long period in June 2005, data from an 84 year old male test subject was recorded and analysed. The results proved to be useful as it showed that human ADLs could be modelled using Markov chains.

In [65] the authors were also interested in designing a person simulation. Their aim was to allow e-Health platforms that need to handle private information of a high number of patients, to evaluate the secure transmission and storage without using real patient information. The authors selected five parameters (Heart Rate, Blood Pressure, Body Temperature, Oxygen Levels, and Respiratory Rate) to be simulated based on discrete events and random normal distribution. The authors planned the simulation of possible deviations from the normal data.

2.2.5 Software Simulation Validation

While the simulations introduced in Section 2.2.4 are an interesting alternative for real laboratory environments, they share the mutual problem of not presenting a mathematical model [63]. It is therefore not possible to simply verify a simulation. Some researchers verified their work with the help of real data that was collected in an AmIE, which cannot be the solution as these simulations are designed to substitute real environments. As long as there is, no 'official' simulation package for AAL related environments other means of validation, verification, and evaluation need to be found. In [66], the authors state that: "Simulation validation in particular is the process of determining the degree to which a model or simulation is an accurate representation of the real world from the perspective of the intended uses of the model or simulation". The

authors carried on to highlight that even when the simulation environment is validated, the simulation result is not necessarily valid for the intended use. Furthermore, as natural systems, like a human being, are complex a reliable validation might not even be possible [67].

When looking at the behaviour simulators described in Section 2.2.4, authors do not validate their model but evaluate their simulation based on the presumption that test subjects will have a consistent daily activity cycle. In the studies of [68]-[70] and [71] it is highlighted that the main target group (elderly) for AAL services show a constant daily routine, as their life is not influenced by work commitments. Moreover, a deviation of such a routine is seen as a possible indicator for a near or current health problem [68]-[70] [6]. In [58], the authors argue that with a simulation environment and enough variation, it is still possible to achieve representative results for AmI services. Furthermore, a possible explanation for the sparse information regarding simulation environments compared to other areas is given in [72]. The authors' opinion is that the strict control of healthcare information in combination with a low number of test subjects and uncertainty of ethical permissions results in a low interest rate of research in this area.

2.3 Monitoring

The German Association For Electrical, Electronic & Information Technologies (VDE) specifies elderly monitoring to be one of the key components of AAL, as it allows the elderly to stay longer in their own home [8]. In [73] the definition of monitoring is given as: "a series of observations over time". Even though the paper is directed at environmental observations, it introduces two monitoring concepts that are interesting for AAL. One is short- and the other long-term monitoring. While for environmental monitoring, the time spans are much wider; the reason behind the two concepts is still suited for AAL.

In short-term monitoring, it is interesting to know if a person can still live independently, while long-term monitoring is more concerned with longer trend.

Based on the EMERGE system [74], short-term monitoring is interested in:

- Do ADLs take place in the correct order, such as is hot water prepared and used for tea, are numbers dialled correctly?

- Is a standard level of hygiene maintained?
- Is medication taken on time and with the right dose?
- Vital signs taken on time?
- Is the person still able to walk properly?

While long-term monitoring is more interested in:

- Does the toilet use increase?
- Abnormality Detection (are vital signs outside their range, are unusually high acceleration values for the person recorded)?
- Did the walking gait change?

The following sections discuss the work found in short- and long-term monitoring.

2.3.1 Short-Term Monitoring

The aim of short-term monitoring is to identify specific events, which can be described a-priori or with the help of expert knowledge. Typical research areas are the classification of gait movements, activity classification, correct temporal execution of ADLs, and data fusion. Each is a complex task but the activities under investigation are limited to a short time frame. It is therefore possible to collect and label extensive amounts of data.

In [75], the authors investigated a biofeedback system based on a tilt angle sensor. The feedback system is designed to improve a subjects' Tandem Romberg standing test results. The test requires the subject to stand heel to toe (shoes removed), arms folded, and eyes closed for 30 seconds [76]. The sensor data is analysed and the user is informed through vibration motors to adjust their posture based on a threshold of the tilt angle.

In [77], the authors concentrated on the step-up and step-down movement of a sit-to-stand transition analysis. Two different data sets were used, the first was collected under laboratory conditions with healthy test subjects, simulating knee impairment, and the second with test subjects in their own home environment after they had a knee operation. The authors discovered that the data collected under laboratory conditions allowed for a better analysis compared to the data collected in real environments. This concurs with the report by [78], where the authors pointed out that the classification accuracy could drop by 1/3 when changing from a laboratory to a test subject's natural environment. Human Activity Recognition (HAR) studies should therefore facilitate a large and diverse group of test subjects, this, while desirable, is mostly not possible. The authors continued to state that another important aspect of HAR systems are the actual

activities investigated, the extracted features, the quality, as well as the quantity of the training set for classifiers, and the classification algorithm itself. It is therefore not surprising that several different HAR systems exist, each one based on different aspects and parameters. Another aspect the authors noted is that HAR studies should be divided into two categories based on the activities investigated. The first category covers the more general ADLs (such as walking, standing, and sitting), while the other studies are interested in abnormal activities (such as a fall).

The work presented in [79] belongs to the first category. The authors combine motion information (tri-axial accelerometer placed on the right thigh) with optical position tracking to recognize eight activities; sitting, standing, lying, walking, sit-to-stand, stand-to-sit, lie-to-sit, and sit-to-lie. Motion sensor data is analysed with a combination of neural network and Hidden Markov Model (HMM). This outcome is then fused with the position information using Bayes' theorem. The authors collected five training and fifteen tests sets for the supervised classifiers under laboratory conditions. They then compared the classification accuracy of motion only data, with motion data fused with position data. The former achieved a minimum accuracy of 65%, while the addition of the position information increased the minimum classification accuracy to 81%.

Another similar study is presented in [80]. The Data Acquisition Device (DAD) used includes an accelerometer, gyroscope, and magnetometer. Placed on the wrist of a subject's dominant arm (left or right), it captures the arm posture, heading, and motion frequency. Furthermore, ceiling-mounted cameras supplement the system to provide the user's position in the environment. The activities of interest are: cooking, eating, brushing teeth, using toilet (up), using toilet (down), cleaning, fetching water, taking out trash. Especially activities such as cooking can significantly vary in time. The preparation of a toast or a 3-course meal can both be considered to be cooking, even though the time may vary from five minutes to two hours. Individual activities are detected using a set of Finite State Machines (FSM), which are initially set up by a field expert. The input parameters for these FSMs are frequency, tilt, heading, locations, and speed, all calculated from the attached DAD and overhead cameras. For the tested activities, the authors achieve 92% and 93% for precision and recall respectively. Precision is thereby a parameter that determines how often the predicted class label for a data segment matched the actual class label. The recall measures is an indication of how many data

segments labelled as a specific class, were correctly predicted to belong to this class. These parameters are discussed in more detail in Chapter 6.

While the two earlier works highlighted more "high level" activities, [55] concentrated on fine-grained activity recognition. The finer activities under investigation are: using the bathroom, making oatmeal, making soft-boiled eggs, preparing orange juice, making coffee, making tea, making or answering a phone call, taking out the trash, setting the table, eating breakfast, clearing the table. The data set was collected using two gloves fitted with Radio Frequency IDentificiation (RFID) readers. The test environment was equipped with 60 RFID tags attached to the different objects used. The complexity of the classification task increases when activities that share objects and activities are executed in parallel instead of a sequential manner. Activity classification with Hidden Markov Models (HMM) achieves an accuracy of up to 88%. Even though fine-grained HAR is of interest, the placement of RFID tags for each object in a room is obtrusive and difficult to achieve.

A different fine-grained HAR is the emergency detection of falls. Several research groups tried to solve the problem using either threshold based solutions or supervised classifiers. In [81], the authors present a fall detection system based on two thresholds for a tri-axial accelerometer that is mounted on the test subject's torso. The authors considered four different falls: forward fall, backward fall, left, and right side fall. Additionally, six further ADLs were recorded: sit-to-stand, stand-to-sit, sit-to-lie, lie-to-sit, bend down to pick up an object, and walk for 2 m. Their system looks at 1.5 s of data with 50% overlap. For each data segment the Signal Vector Magnitude (SVM) is calculated. SVM is a measure of movement intensity and is calculated using equation 2-1, with $x, y, and\ z$ being the three different axes of the accelerometer and i an element of the window segment.

$$SVM_i = \sqrt{{x_i}^2 + {y_i}^2 + z_i^2} \tag{2-1}$$

The developed algorithm classifies the individual data segments based on the containing SVM values. For a window segment to be classified as a fall it needs to include a minimum SVM value of at least 0.75 g and a maximum value of above 1.75 g. The authors state that to reduce false positive fall classifications, it is necessary that the lower limit is passed before the higher limit is reached. For all other conditions the algorithm predicts the window segment to be part of a normal ADL. The achievable sensitivity and

specificity were 100% and 93.33% respectively. Sensitivity is interchangeable with the earlier introduced recall measure and also indicates how many data segments labelled as true, were correctly predicted to belong to this class. Specificity is similar but instead of determining the rate of correctly classified true data segments, it is used to determine the correct false classification.

Another system that is based on threshold values and accelerometer sensor data is presented in [82]. The sensor is mounted to the waist of the test subject to record movement information. Besides fall events, the authors recorded the following activities to test their system's sensitivity and specificity: sitting, standing, lying, walking (slow and fast), going upstairs, going downstairs, sit-to-stand, stand-to-sit, stand-to-lying, lying-to-stand. The authors use raw accelerometer data for the Y-axis and unspecified lower and upper thresholds.

A different approach is presented in [83]. Instead of using threshold values, a combination of fuzzy logic and artificial neural networks evaluates the accelerometer data of a test subject. In the first stage, the sensor data is transformed into spherical coordinates, which is followed by the calculation of the Signal Magnitude Area (SMA). This information is fed into a neuro-fuzzy system. The output is compared to a threshold and when exceeded twice in a row, classified as a fall. The system was tested on four typical falls: collapse, forward, backward, sideward. The achieved sensitivity and specificity were at least 88% and 99.5%, respectively.

In [84], the authors described two independent fall detection algorithms that can also be used in cooperation with each other. The first system is based on a camera in combination with Particle Swarm Optimization and the second one on a tri-axial accelerometer using Hierarchical Temporal Memory (HTM). The first algorithm is designed to detect a moving person using a camera. The algorithm marks the outline of the person and determines the current position, standing, bending, and lying on the floor. A fall is detected when the outline of the moving person changes from height > width, to height < width. The second algorithm works without a-priori knowledge. The HTM algorithm is combined with an SVM classifier to learn and generalise the temporal patterns of the underlying accelerometer data. The combined system is trained with a training set of ten and a test set of twenty events per activity. The authors do not highlight the classification accuracy for the particle algorithm but the HTM algorithm achieved an accuracy of 97.05%.

2.3.2 Long-Term Monitoring

The preceding section showed that several works investigated fall detection as part of abnormal activity using a-priori knowledge. These short but current time frames allow the inference of a person's current health state, where good health is represented by normal activity. The problem though is that the work presented above was looking into specific ADLs. Even though, this is a legitimate approach, additional information about a person's health can be gained from looking at long-term data. While a human is not a machine, literature still shows that deviations in certain parameters established over a longer period of time can highlight unspecified current or near health problem. Therefore, this section will discuss literature that presents the effectiveness of long-term data monitoring.

In [69], the authors introduced the Circadian Activity Rhythm (CAR). The work stated that the human behaviour is influenced by a social (hard to measure) and a biological (measurable) rhythm. Especially elderly are mostly driven by the biological rhythm, as their social interaction reduces over time. The authors validated the CAR rhythm with data from 22 residential care homes. Additionally, the data from three test subjects were automatically analysed for changes in the CAR rhythm. Deviations resulted in alarms, which were discussed with the responsible care personnel. The authors highlighted that every deviation automatically has its source in behaviour abnormality. For one of their test subjects, the daily deviation had to exceed 25 alarms before being considered abnormal behaviour.

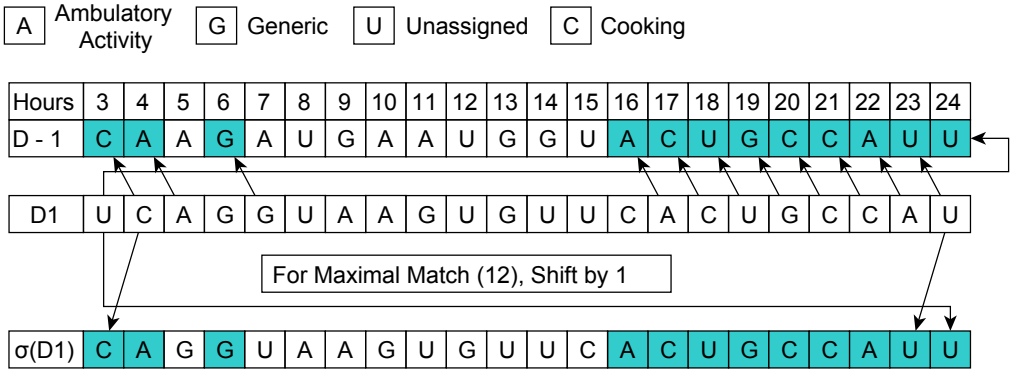


Figure 2–2: Explanation of nycthemeral shift (Adapted from [85])

In [85] and [86], the authors investigated the detection of a nycthemeral shift, which in this context is a change in activities for consecutive days, as they state its usefulness as an indicator for a rapid decline in wellbeing and a possible early warning sign for dementia related problems. For their work, the authors divided days in 24 one-hour slots

and 96 quarterly-hour slots. Each timeslot is defined by the majority activity that was observed during the time period. The six activities under investigation are: outside, ambulatory activities (between rooms at home), generic social or cultural (reading, watching TV, receiving guests), unassigned (resting or sleeping), personal hygiene (dedicated activities), and cooking and eating (dedicated activities). Figure 2–2 illustrates the activity comparison between two consecutive days (D - 1 and D1). The maximum number of activity matches between the two days is twelve, with a nycthemeral shift of 1 for D1. In case of insufficient matches, an alarm will be raised.

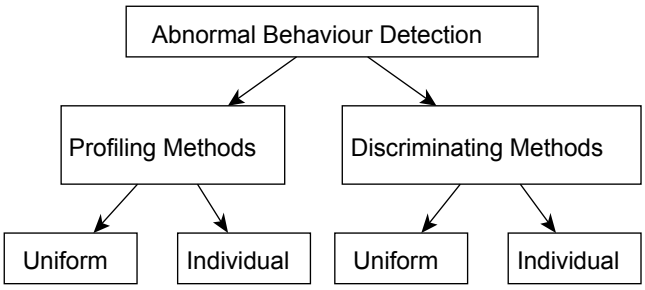


Figure 2–3: Classification approaches for the abnormal behaviour detection (Adapted from [87])

In [87], several interesting statements are made. The first part of their work states that there exist two different methods (profiling and discriminating) for the abnormal behaviour detection, which are represented in Figure 2–3. The former requires only normal activities in the training set and can detect a deviation of the learned model, which is classed as abnormal. The latter requires the training set to contain normal and abnormal activities to establish two independent data models. Each method can be further separated into a uniform and an individual approach. The uniform approach tries to generalize, e.g. all monitoring subjects behave in a similar way, while the individual approach aims to establish separate models for each monitoring subject. The second part described their approach of combining three sub systems (regression analysis for circadian rhythms, histogram-based analysis for movement data, and a probabilistic model based on the persons' ADL) to improve abnormality detection. One of the main components in the system is an Activity Recognizer (AR) component, which processes sensor data based on a combination of AI algorithms e.g. case-based reasoning, rule-based reasoning, time maps, and fuzzy approaches, to extract ADLs. Their system is the most sophisticated approach the author of this thesis has come across.

In [11], the University of Kaiserslautern presented their work to equip 20 flats with AmI technology for a joint research project with the aim to provide monitoring services such as comfort, safety and health. With the help of 30 different sensors (PIR, door and

window reed switch, water flow sensor etc.) the group generated activity/inactivity models of the test subjects.

In [88], a behavioural pattern monitoring system is described based on eight motion sensors (bedroom, bathroom, office, living room, kitchen, laundry room, front door, and shower). Sensor data includes location, start time, duration, and as an activity parameter they divide the number of consecutive sensor activations by the duration. The authors assume that sensor data is the result of different activities carried out by the test subject. Therefore, a mixed model approach is used to cluster the sensor data into different observation clusters. The authors used 40 days of sensor data as the training set for the mixture-model. The data was grouped in 139 different clusters of which 44 were significant. Significant in this context means that the cluster is identified in at least 25% of the days or that it met a length or time-range criteria. The authors concluded that significant clusters can accurately identify activities such as sleep behaviour, bathroom usage, and changing clothes.

In [89] the idea was presented to use Naïve Bayes as a supervised learning method. The system is required to allow for online learning, so the learned model should adjust to changes in the person's behaviour. The authors continue to state that supervised learning is required, as there exists many layouts of homes and unique ways to perform an ADL. The use of supervised classifiers required labelled sensor data, which was pointed out as a problem by the authors. Therefore, test subjects were equipped with Personal Digital Assistants (PDAs) to label their own sensor data. However, the generated data was not sufficient for the training of the classifiers. This required additional time by the authors and subjects for the labelling of still unknown data. As a possible reason for the poor quality of labels, the authors identified the frustration the subjects experienced with the PDA. The initial labelling of new activities was not a problem but subjects found the repetitive labelling of previously known activities irritating. Moreover, the authors pointed out that there was a significant variation in labelling identical activities, which raises the question of how efficient and necessary labelling of ADLs for abnormality detection is in the first place.

A case-driven AmI (C-AmI) system is presented in [90]. The authors present a system that can handle multiple cases discovered in an AAL environment to allow for the best possible care of an elderly person. While this particular work only highlighted ADL classification based on a rule-based system, the authors also criticized the dull task of

acquiring a-priori knowledge for AAL research. Moreover, the work stated that activity recognition becomes more a field of data mining techniques and emphasize that rule based AI methods will have problems with the complex system. This is emphasised by the 820 to 920 rules generated for the 20 different ADLs under investigation. Even more rules could be required when ADLs are not mutually exclusive and can happen in parallel. The authors state that at this point, typical static classification methods such as Naïve Bayes, decision trees, and K-Nearest Neighbour (KNN) might be preferred.

In [91], the authors stated that behaviour is defined as any pattern in a sequence of observations. Their work centred on identifying small behaviour changes and abnormal behaviour where the abnormal behaviour can originate either from a sensor failure or an actual deviation in behaviour by the test subject. The authors state that their approach can detect deviation when using temperature and light sensors but cannot distinguish between a sensor failure and a human behaviour deviation.

In [92], the authors presented a context-aware system that fuses the output of two different pressure sensors to detect a person's health change. The system generates a set of normal rules for the test subject and compares the daily morning routine of getting out of bed and going to the toilet with past information. The first routine is monitored using a pressure mattress under the subject's bed, while the second routine is controlled using a pressure sensitive grab bar attached to the toilet commode. The system has three warning stages: OK, Warning, and High Warning for both routines. Having multiple warning stages for multiple activities allows the user to fail one, while passing the other without causing false alarms. The authors do not highlight how the initial baseline information is gathered. The authors reported a wakeup time of 7 am for one user, but do not highlight how this is determined and how the system would react to e.g. a 15-minute delay in the wake up routine.

In [93], the authors criticised the limited pre-selection of ADLs in the classification literature for AAL, which in their view, is a restricting factor for wellbeing monitoring and assistance systems. Therefore, the authors designed a system that is able to detect frequent events in the subject's daily routine and track the occurrence of such events. This allowed the system to not only monitor the subject's wellbeing but also notice possible behaviour changes. The authors used their own Activity Discovery Method (ADM) based on their own sequence-mining algorithm (Discontinuous Varied-Order Sequential Miner (DVSM)) and clustering algorithms to identify frequent sensor events

that most likely belong together. Individual HMMs and a voting multi-HMM for the final activity classification are trained with the discovered activity sequences. The authors tested their ADM algorithm with 20 test subjects executing five different activities (telephone use, hand washing, meal preparation, eating and medication use, and cleaning). The results showed that the ADM algorithm managed to cluster 80% of the activities and 87.5% of individual sensor events were assigned to the correct activity cluster.

Another approach to long-term monitoring is presented in [94]. The authors of this work pursued exactly the opposite direction of [93]. Instead of grouping frequent activities, they tried to identify rare and unexpected events. The authors' reasoning for the interest in anomaly detection in Ambient Intelligence Environments (AmIEs) is the possible identification of situations of importance (e.g. a fall), improving the collected data for activity recognition (reducing noise), as well as contacting the inhabitant in these abnormal situations. The authors note that the anomaly detection in AAL is a relatively new research direction and simple classification rules set up by experts, will not be sufficient for AmIEs. This is in agreement with [90]. Therefore their approach is to use a One Class Support Vector Machine (OCSVM) for an initial experiment using past data from their smart home MavHome. The classification algorithm was trained using normal data and in the test data presented with abnormal events. While the authors fail to mention what exactly constitutes to an anomaly in this context the work reports a precision value of 100%, a recall value of 100%, as well as an f-measure value of 100%.

2.4 Artificial Immune System

Most of the literature presented above has reported work, which has tried to locate shifts in recurring data patterns that represent abnormal behaviour patterns or unknown abnormal data points in an otherwise unknown and unlabelled data stream. In [87], the authors drew parallels to Intrusion Detection Systems (IDS) or fault detection. In [95], the authors gave an overview of IDS found in the literature. The main focus point of the authors was Anomaly Based detection Systems (ABS). The advantage of an ABS is the ability to detect unknown intrusions (also called Zero-Day). The disadvantage is the complexity of such systems and that even standard attacks require some learning time. One of the authors' introduced methods for ABS is based on Computer Immunology (CI). It is presented that the Artificial Immune System (AIS) allows for an adaptive system,

which has been used in research areas, such as virus detection, decision support systems, network intruders, fault detection etc. However, it is interesting to note that neither literature regarding detection of abnormality in Ambient Assisted Living or AIS in general mentioned the combination of the two fields. The only reference the author found was [96], which was presented at a conference after the author suggested the use of AIS for fall detection in AAL in [97]. Furthermore, the work presented in [96] uses AIS to cluster the outcome of a fall risk analysis questionnaire and not a data stream of an accelerometer sensor as presented in this work. For the author it is unclear why AIS is not wider used in AAL, as also no negative experiences has been found. Especially since the short- and long-term monitoring sections above highlighted the requirement of pattern classification, detection of novel patterns, changes in the subject's behaviour, and complete sets of normal and abnormal behaviour are not available. This fits with [98], where it is stated that AIS works best in applications that require pattern classification and responses, a distributed architecture, the detection of novel anomalous patterns, system behaviour changes slowly over time, and only limited storage is available that does not allow to store the full normal system pattern. The author believes that AIS might be an overlooked method for AAL and its use for this important field is worth investigating. The focus of this section is the introduction of AIS, present and past work, and why it is suitable for abnormality detection in AAL. The functional principles of AIS will be covered in detail in Chapter 4.

2.4.1 Introduction to Artificial Immune System

As a general overview, AIS is an Artificial Intelligence (AI) technique, which is an algorithmic representation of the Human Immune System (HIS). The HIS is a system of biological structures and processes that work to prevent diseases in humans, through the detection of harmful agents (foreign bodies). Therefore, human body cells need to be divided into a self and non-self set (in terms of AAL normal and abnormal behaviour patterns) using the chemical surface of a body cell to determine if a cell has been discovered before or not. Under normal circumstances, every human is born with a functional immune system, ignoring immune deficiencies or allergies. While the HIS of a baby is at its initial stage, it allows for the early protection outside the mother's womb. Over time, the HIS will be exposed to several unknown agents (harmful and harmless). It will react to them (immune response), learn, and adapt because of these encounters. It

follows that the HIS is individually tailored to a single person, evolves over time (online learning), and is initially trained from normal patterns without remembering every single cell or being supervised [98], [99]. Different researchers have shown that the characteristics of the HIS can be modelled using AI terminologies to work outside the human body. While AIS can be used for pure optimisation problems, [100] suggests that AIS, because of its root in evolutionary algorithms, is best suited for dynamic problems, which change over time. The authors continue to list the advantages of AIS to be: distributed, adaptive, self-organising, and online learnable. It is therefore used in a wide field in research [95], which share the common interest to "find" the normal self-set of the computer network, machine or data set and trigger an alarm in case of a deviation from the norm.

2.4.2 Related Work

In [98], the authors designed a general purpose architecture for AIS and proofed their concept using a Lightweight Intrusion detection SYStem (LISYS). LISYS was evaluated using simulated network data to generate a normal model. The system was then tested with seven unknown attacks, which were all detected by LISYS. Furthermore, the authors point out that the use of a detector activation threshold achieved a low false positive rate without significantly decreasing the attack detection rate (true positive).

In [101], a similar IDS based on AIS is suggested. The presented work was just in the initial stage and no progress on the detection accuracy was reported. While this work was only in its early stage, another network anomaly detection system is presented in [102], where the authors combined AIS and fuzzy networks. Results highlighted a good detection rate of AIS for real network traffic.

Besides IDS, AIS can also be used for Intrusion Prevention Systems (IPS) as shown in [103]. The main components of the presented IPS are: SEnse detection Agent (SEA), ANalysis Agent (ANA), ADaptive Agent (ADA), and Self-Healing Agent (SHA). The ANA and ADA are based on AIS algorithms. The presented paper is only in the preposition stage and the authors provide no verification.

In the area of fault detection, the authors in [104] compared the detection performance of Artificial Neural Networks (ANN), Logistic Regression (LR), Support Vector Machines (SVM) and K-Nearest Neighbour (KNN) to Artificial Immune Recognition Systems (AIRS). The authors used the different machine-learning algorithm to classify real fault data

(power distribution) from three regions in North Carolina. The results showed that the AIS-based algorithm outperformed the other tested algorithms most of the time.

A different work which investigated the use of AIS for fault detection is reported in [105]. The authors described predicting all possible faults for a multi-robot system as a complex task because faults could happen in the robot hardware or even in the programming. They therefore introduced AIS as an anomaly detection algorithm in each robot to monitor the surrounding robots instead of the individual system. An individual robot was determined to be faulty, when the majority of the surrounding robots detected abnormal movement behaviour. This work did not only highlight the abnormality detection ability but also made use of the possible distribution features of AIS. The system was tested using simulated robots, with different motion behaviours and faults. Depending on the robot movement and the simulated fault, the algorithm achieved good results (up to 100% detection rate).

In [106], the authors attributed the increased complexity of electronic systems (hardware and software) to the difficulty in fault diagnosis of these systems. Today's electronic have to cope with material defects, components of varying quality, and diverse application ranges. It is therefore near impossible to anticipate all possible faults. The authors point out that electronic units, which pass a list of a-priori fault test routines are automatically assumed to be in good working condition. Therefore a fault detection system is proposed based on AIS to detect unknown faults and their location in a specific subsystem that is responsible for the abnormal data in a complex system. For the validation, the authors used software simulations of a diesel engine Exhaust Gas Recirculation (EGR) system and generator portion of a commercially available marine diesel-generator system. Each simulation environment consists of several smaller sub systems, which were used to inject faults. Unfortunately, the authors do not name the fault detection rate besides stating that faults are detected and localised using their algorithm.

A different but also interesting use of AIS is shown in [107], where a subject's domestic heating behaviour was learned and a AmIE's heating patterns adjusted accordingly. AIS was used to act quickly on the user's special heating demands without forgetting its normal behaviour. The authors tested the system with the scenarios of regular use, the regular use with a break of two weeks of no use, the regular use with a change of use

after a number of weeks and irregular use with frequent changes. The AIS was able to adapt and could still react to changes in the user's behaviour.

In medical research, AIS scored especially well in [108]. The authors classified ElectroEncephaloGraphy (EEG) and lung cancer data sets using AIS. Their results showed an achievable accuracy of 100% on the data sets and pointed out that this accuracy is the highest among the classifiers reported in the literature. One reason for these good results is attributed to the self-learning ability of AIS since it allows for a fully automated classification.

In [109] the authors propose to use the AIS paradigm in the area of Ambient Intelligence (AmI), specifically for facial expression detection with additional context information. The authors also investigated human behaviour recognition but did not make any connection to health related monitoring.

2.5 Summary

The literature review covered available AAL related data sets, hardware designs of DADs, AAL related simulation environments, and the on-going research for short- and long-term ADL classification and abnormality detection. During the review, the Artificial Immune System (AIS) was discovered as an Immune Intelligence method and suggested by the author as a novel method for the unsupervised long-term monitoring, to identify abnormal movement of an elderly to detect falls without prior training.

As the literature did not highlight any useful data set that include human falls and the information regarding the identified DADs is not enough for the reproduction of such a device, an Open Hardware Data Acquisition Device (OHDAD) has to be developed for the collection of normal and abnormal human activity. The data collected by the OHDAD is used in combination with a Markov chain-based simulation environment for the generation of various long-term activity scenarios, which will be the basis for work carried out in this thesis. The primary gap discovered in the literature review is that in the field of AAL, AIS is not used for the detection of abnormal activity. Introducing AIS to the field of fall detection in AAL and closing this gap is not only beneficial for AAL related research but also increases the AIS related knowledge with regards to its usefulness in other non-related research subjects. The literature also highlighted an inconsistency in parameter combinations for the different supervised classification methods used in AAL. In order to select fair parameter combinations for the comparison of supervised

classifiers with AIS, part of the thesis is therefore considered with the influence of parameters and their combination on the classification accuracy and computational load. The value of this literature review is to demonstrate the feasibility of adapting AIS to the proposed unsupervised abnormality detection and its usefulness there.

3 An Open Hardware Data Acquisition Device and Software Simulation Environment for Long-Term Data Generation

3.1 Introduction

This chapter presents a new Open Hardware Data Acquisition Device (OHDAD) for the collection of 10 Degrees of Freedom (DoF) sensor information and a new long-term data simulator environment, which were both developed as part of the work presented here. The former was created after literature only highlighted Data Acquisition Devices (DADs) that could not be reproduced, as no schematics are available, were based on out-dated hardware, or did not provided the full functionality required for the work presented in this thesis. In addition, a simulation environment based on Markov chains is designed to accompany the OHDAD. The simulation environment and short-term data, collected with the OHDAD, allow for the fast generation of long-term data, which can be used for the abnormal activity detection. Both elements proved to be vital components as they have enabled data collection, generation of well-defined scenarios, and reduction of collection and data labelling time.

Section 3.2 discusses in detail the OHDAD's design requirements, electronic component selection, design approach, and validation of hardware functionality. The results of this section were also presented in [110]. Section 3.3 presents an introduction to Markov chains, the design of the simulation environment, the underlying Markov chains, the selection of transition probabilities, and an evaluation approach. Part of this work was also presented in the author's peer-reviewed conference publication [111]. Section 3.4 summarises the chapter.

3.2 A New Open Hardware Data Acquisition Device (OHDAD)

3.2.1 Motivation for the Design of a Data Acquisition Device

A data set containing normal Activities of Daily Living (ADL) and abnormal activities (e.g. a fall) is required when the detection of abnormal activity is of interest. The data sets identified in Chapter 2 contained only normal activities. It was therefore necessary to collect new data sets that include both types of activities.

The Data Acquisition Devices (DADs) discussed in the literature review and required for the data collection, were designed for specific experiments and the hardware

information were not detailed enough to enable reproduction. The only DAD that was described in enough detail is the Hoarder board from the Michigan Institute of Technology (MIT) but is based on out-dated hardware components [34].

Furthermore, two commercially available hardware systems were tested, which in the end were both considered not fit for purpose. The first system tested was a Real-Time Location System (RTLS) from UbiSense [112], which resulted in similar observations as to [26]. For a location reporting system, an accuracy of ±15 cm can mean the difference between sitting on a chair and lying on the floor. The main drawbacks of the system are:

- Only usable indoors
- Position accuracy depends on accurate measurements of the used rooms and line of sight
- High initial costs

An inexpensive alternative is the Texas Instrument (TI) development kit eZ430-Chronos [113]. The development kit is designed as a sports watch with an inbuilt tri-axial accelerometer and a wireless ISM module (transmitting on either 433 or 868 MHz). The initial investigation is presented in [114]. The drawback of the system was the limited available internal storage, which required a base station nearby for the sensor data collection. Without a data communication protocol, data could have been unknowingly lost during the data transmission. As no suitable DAD could be found, the author pursued the design and implementation of a DAD to enable the collection of human activity data in- and outside the laboratory environment for short- and long-term behaviour analysis, with the aim of being open, affordable, and simple to reproduce. The openness of the system is one step in the direction of sharing sensor data, which was urged in [7]. The following sections detail the design requirements, electronic component selection, design approach, and validation of the hardware functionality of the OHDAD.

3.2.2 Design Requirements

The requirements for the developed OHDAD, with respect to the DADs highlighted in literature, are the following:

- Small and unobtrusive
- Long-term wearable (at least one day)
- Sensor diversity (for future research and other research groups)

- Integrated Development Environment (IDE) should run on as many OS platforms as possible (at least Windows, Linux, and MacOS)
- Programmable through USB connection
- Wireless connectivity
- Affordable

The block diagram of the OHDAD's hardware architecture, illustrated in Figure 3–1, is derived from the works of [30], [31], [34], [36]. The microcontroller is able to collect sensor information from analogue as well as digital sensors, using either the in-build Analogue-to-Digital Converter (ADC) or the Digital Input/Output (DIO) pins. Local storage allows saving extensive amount of sensor information that is collected over long periods of time. The microcontroller can be programmed from a host PC using the USB connection that doubles as a data communication channel. In addition, a wireless module enables the communication with a nearby base station.

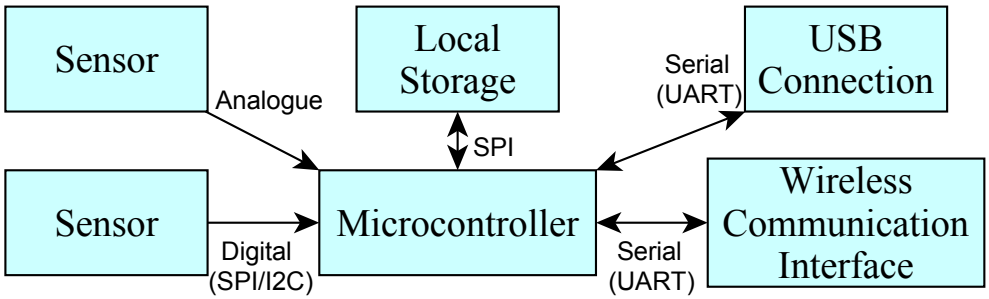


Figure 3–1: Block diagram of the Open Hardware Data Acquisition Device

3.2.3 Components

The following sections outline the different component selections of the microcontroller, sensors, storage, and wireless communication for the OHDAD. The three tri-axial sensors (accelerometer, magnetometer, and gyroscope) and the barometer allow for the recording of 10 Degree of Freedom (DoF). Each sensor reading is time stamped, locally stored, and wirelessly transmitted to a base station for easier data labelling during post processing.

3.2.3.1 Microcontroller

Two microcontroller systems were considered as the base of the monitoring system. The first option is the Arduino platform [115] that comes in various sizes and is based on different Atmel microcontrollers like the ATmega168, ATmega328, and ATmega2560. The second option is the mBed platform, which is based on the ARM Cortex-M3

processor [116]. The latter having the advantage of increased processing power and the disadvantages of a prohibitive price and a larger physical size. In its first stage, the system is designed purely as a data acquisition device. Hence, the factors of size and cost are more important than processing power. Therefore the Arduino Nano V3, based on an ATmega328 was selected. A smaller size could have been achieved with an Arduino Mini, based on the same microcontroller, with the disadvantage of no on-board USB connectivity, which would have required additional hardware for the programming. In addition, the Arduino IDE is written in Java and available for Windows, MacOS, and Linux.

3.2.3.2 Sensors

The literature concentrating on body worn devices, described in the literature review in Chapter 2 is mainly focused on Micro-ElectroMechanical Systems (MEMS) sensor data. The main sensor used for fall detection is an accelerometer (bi- and tri-axial) [31], [36]. It is followed by bi- and tri-axial gyroscopes [117] (or in combination) with accelerometers [118]. In addition, most modern smart phones include at least an accelerometer, with more vendors also including gyroscopes, magnetometers, and even barometers. These smart phones not only allow for a resourceful data collection but also on-board data classification [119], [120]. The problem though is the short battery lifetime when used for such tasks. As the platform described here should offer sensor diversity and provide researchers with various sensor information, it was decided to include at least sensors that could be found in a modern smartphone for a fraction of the price and prolonged battery lifetime. The total cost for the hardware presented here is below £100, which, for now, is cheaper than a smartphone with similar sensors. In addition, algorithms used as part of the work presented here could be implemented on smart phones having similar hardware specifications.

Table 3–1: Hardware specifications of the sensor boards used for the Open Hardware Data Acquisition Device

	9 DoF Sensor Board (SEN-10724) from Sparkfun			SEN-11282 from Sparkfun	ChronoDot from Macetech
	Accelerometer (ADXL345)	Gyroscope (ITG-3200)	Magnetometer (HMC5883L)	Barometer (BMP085)	Real Time Clock (DS3231SN)
Power Supply	2.0 V - 3.6 V	2.1 V - 3.6 V	2.16 V - 3.6 V	1.62 V - 3.6 V	3.3 V
Power Consumption (Measuring)	40 µA	6.5 mA	100 µA	5 µA at 1 sample/sec	200 µA
Sensor I/O Interface	SPI / I2C	I2C		I2C	I2C
Number of Axes / Measurement	3			N/A	Seconds, Minutes, Hours, Day, Data, Month and Year with Leap Year Compensation Valid Up to 2100
Sensor Resolution	10 Bit	16 Bit	12 Bit	16 to 19 Bit	N/A
Available Range	±2 / ±4 / ±8 / ±16 g	±2000 ˚/sec	±8 Oe	-500 m to 9000 m	N/A
Sensitivity	4 mg/LSB	14.375 LSBs per º/sec	2 mG Field Resolution	0.25 m	±3.5 ppm

MEMS sensors are only available as very small Surface Mounted Device (SMD) components that hinder their use in non-commercial environments. Therefore, several companies started to produce breakout boards to convert these sensors to Throughput Hole Technology (THT) components, which can be easily used by the open source community or in commercial environments for rapid prototyping projects. Another advantage of using breakout boards is their wide availability, the large support community for software libraries, low cost and distribution of schematics under the open hardware license. The disadvantage of breakout boards is the increased component size; it is therefore preferred to use boards that combine several MEMS sensors. The three main breakout boards used for OHDAD, with the respective specifications can be found in Table 3–1.

The first breakout board combines an accelerometer (Analog Devices ADXL345 [121]), gyroscope (Invensense ITG-3200 [122]) and magnetometer (Honeywell HMC5883L [123]). The other two boards include a Bosch BMP085 [124] and the Maxim DS3231SN [125] for the on-board barometer and Real-Time Clock (RTC) respectively. The barometer provides the atmospheric pressure and the temperature of the sensor itself. From the former it is possible to gain altitude information that has been used in fall

detection [126], [127]. In [36], the authors highlight the use of data time stamping for received data at a base station to improve the contextual information of the data itself. The OHDAD is equipped with a RTC to reliable time stamp collected sensor data over a long period of time. This enables the system to collect data outside a controlled laboratory environment without a base station in range. All five sensors can be powered using a 3.3 V supply voltage source, which is directly available from the microcontroller platform. The required current varies from 5 µA during active operation (1 sample per 1 second) for the barometer to 6.5 mA for the gyroscope. Furthermore, all sensors used the digital two-wire I2C communication protocol, allowing up to six additional analogue sensors and further digital sensors using the I2C protocol to be attached. The number of digital sensors is only limited by the number of available bus addresses. Another advantage of using digital sensors is the reduced noise on the sensor data itself, as the analogue to digital conversion is carried out on the MEMS sensor as opposed to the microcontroller, where the analogue signal is exposed to signal noise for longer periods of time.

3.2.3.3 Local Storage

The internal EEPROM storage of the ATmega328 is limited to 1 KB, which is insufficient for long-term data collection. The two methods found in literature are to transmit data to a base station [32], [36] or store the information on a local flash storage [34]. The downside of the former approach is two-fold; there needs to be a base station in range and data can be lost during transmission. The flash storage is in the author's opinion preferable and implemented based on a micro SD card (flash storage). In addition, the three main operating systems (Windows, Mac OS, and Linux) can read the micro SD card, providing adequate formatting using the File Allocation Table (FAT) file system. Data access is realised over the SPI protocol, with Figure 3–2 illustrating the pin configuration for the SD card. In this way, sensor data can be collected outside a controlled laboratory environment, saved to a file on the removable storage, and transferred at the end of the experiment from the hardware platform to a PC for post-processing.

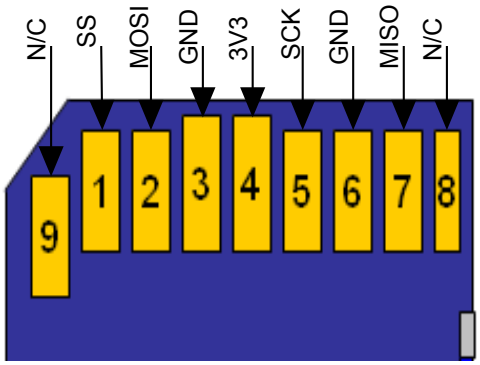


Figure 3–2: SD card pin configuration

3.2.3.4 Wireless Communication

For meaningful data collection, the collected sensor information needs to be correlated to Activities of Daily Living (ADL) labels. This is best achieved with a logbook that is used to record the times of the execution of different ADLs. To match the correct ADL label to the sensor data during the later post-processing stage, the on-board RTC and the clock used for the recording need to be synchronised. Therefore a wireless communication interface is included in the OHDAD to allow the inspection of correct operation of the platform, observation of the time stamp of the current sensor reading, as well as the transmission of the sensor reading for direct processing at a base station. Several radio options are available, based on technologies like ISM, WiFi, ZigBee, and Bluetooth (BT). Most modern PCs are stock equipped with WiFi and BT, and using an adapter board ZigBee communication is possible as well. In the white paper [128] the different technologies ZigBee, WiFi, and BT are compared and their advantages based on the application field pointed out (see Table 3–2). The required bandwidth for the transmission of the collected sensor data is low and therefore all three technologies are adequate in that sense.

Table 3–2: Comparison of ZigBee, WiFi, and Bluetooth (Source: [128])

	ZigBee	WiFi	Bluetooth
Battery Life	Years	1 Week	1 Week
Bandwidth	250 Kbps	Up to 108 Mbps	720 Kbps
Advantages	Low Power	Ubiquity	Convenience

The two important criteria for the selection of communication technology used for OHDAD are battery life and connection ability. In terms of battery life, the ZigBee technology is the most favourable, while BT is the most convenient choice, as it allows for a simple connectivity to PCs and smartphones. The initial design bases the communication on the latter for the convenience to connect to a smartphone. In case

lower power consumption is necessary at a later design stage, a BT module with the ZigBee footprint is used for the OHDAD platform.

Table 3–3: Hardware specifications of the Bluetooth wireless communication module

	Radio Chip	Bluetooth Specification	Frequency	Power Supply	Power Consumption	Communication Range	Maximum Serial Baud Rate
Bluetooth Module	BC417143	V2.0 + EDR	2.4 GHz ISM Band	3.3 V	50 mA	10 m	1382.4 Kbps

The specifications for the BT module are listed in Table 3–3. Its main advantages are the pin compatibility with ZigBee modules and the requirement of a 3.3 V power supply, as necessary by the other sensors presented above. The 10 m communication range allows test subjects to have a certain degree of freedom (movement wise), when performing tasks under laboratory conditions with another person wirelessly monitoring start and end times of an activity on the OHDAD.

3.2.4 Open Hardware Design Approach

The component selection above led to two different design approaches that were considered for the OHDAD. The first one was to design the complete system using a single Printed Circuit Board (PCB) with SMD components to achieve a comparatively small footprint. This would have required the necessary equipment for SMD soldering and advanced soldering skills, hence limiting the audience for the OHDAD. The second one, and preferred by the author, was to incorporate the different breakout boards in the design process. While increasing the platform size, this approach simplifies the reproducibility for others. The schematic (see Appendix A) and the resulting 2-layer PCB (see Figure 3–3) were designed in CadSoft Eagle 6.2.0 [129].

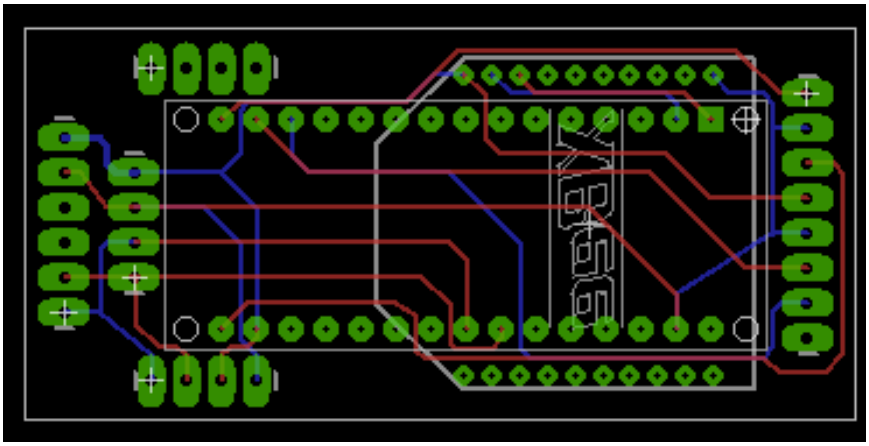


Figure 3–3: Top and bottom circuit board layout of the Open Hardware Data Acquisition Device

The footprint of the OHDAD circuit board measures 60 mm by 29 mm and was produced using a LPKF rapid prototyping machine [130]. Figure 3–4 illustrates the attempt to

achieve a small footprint through stacking sensors on top of each other. The top site is holding the microcontroller, and stacked above is the SD card and RTC. The bottom site holds the 9 DoF sensor board, with the BT module and barometer placed above. The total height of the board is 41 mm with a weight of 38.9 g without battery pack. The software running on the Arduino Nano platform, was developed using the Arduino's own Integrated Development Environment (IDE), with the flow diagrams for the sensor initialisation and data collection illustrated in Appendix B and Appendix C, respectively. All necessary files required to reproduce the OHDAD can be found in [131].

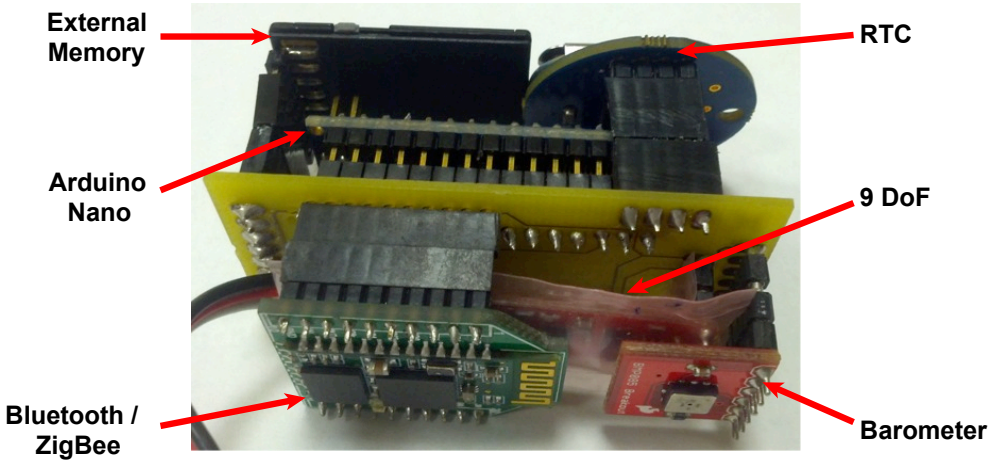


Figure 3–4: Fully assembled Open Hardware Data Acquisition Device

3.2.5 Hardware Validation and Data Collection Evaluation

After the hardware design, production, and assembly, the correct function of the OHDAD and its accompanying software running on the Arduino has to be verified. The OHDAD stores collected sensor data following the Request For Comment (RFC) standard 4180 (see RFC4180 [132]), which is designed for the exchange of tabular data between spreadsheet programs. The sensor reading is stored as plain text in a comma separated value (CSV) file on the external flash storage. Each data recoding is appended to the end of the file and can be viewed with Matlab [133] or a spreadsheet program for further processing. Each recording consists of 18 sensor readings, that are in the following order: year, month, date, hour, minute, second, scaled accelerometer X-Axis, scaled accelerometer Y-Axis, scaled accelerometer Z-Axis, scaled magnetometer X-Axis, scaled magnetometer Y-Axis, scaled magnetometer Z-Axis, gyroscope X-Axis, gyroscope Y-Axis, gyroscope Z-Axis, temperature, air pressure, and altitude Above Mean Sea Level (AMSL).

The OHDAD's dimensions are small enough to be usable at various body locations without being too intrusive. Possible locations are right hip, dominant wrist, non-dominant upper arm, dominant ankle, and non-dominant thigh [25], [134], with [28] suggesting the waist as an adequate position. Therefore, the test subject was wearing the hardware platform in a plastic casing (100 mm x 65 mm x 36 mm) attached to the subject's belt using a Velcro band (see Figure 3–5). A smaller version of OHDAD could be even sewn into a belt.

Figure 3–5: On-body position of the Open Hardware Data Acquisition Device

An initial test trial is performed to validate the correct working order of the OHDAD and to determine the achievable sampling frequency. The test entailed the test subject using a set of stairs in the institute to descend two levels, passing a corridor and using another set of stairs at the end of the corridor to get back to the second level, walk back to the office and sit down in an office chair. From the four available sensors, the visual analysis is concentrating on the accelerometer, as it is a core component of the work presented here. In addition from the three remaining sensors, the barometer is the best information source to present the movement through the institute. The visualisation of these two sensors represents the correct working of the two main sensor boards and hence the correct working of the OHDAD components (hardware and software).

The accelerometer output is illustrated in Figure 3–6. The data shows the constant movement of the test subject with a stationary phase at the end when the subject reached the chair and stopped producing any significant movement.

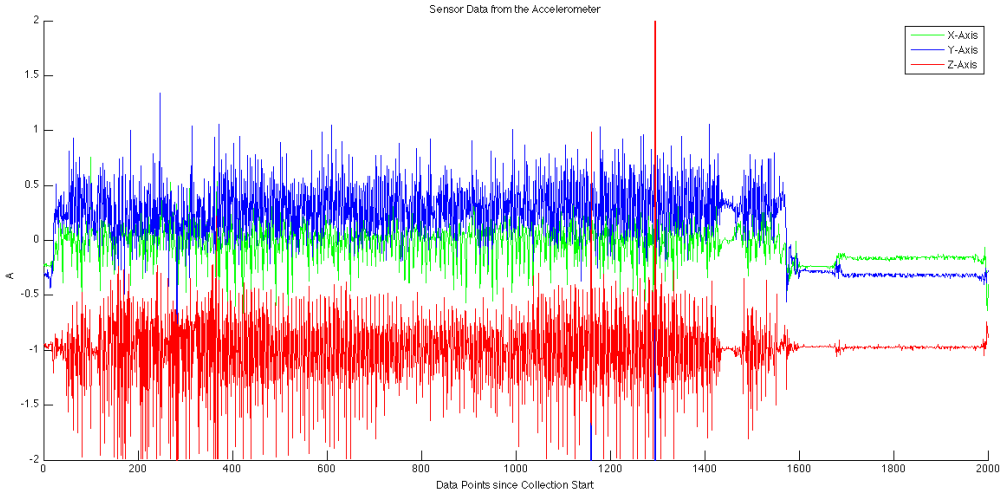


Figure 3–6: Accelerometer sensor output

Figure 3–7 illustrates the same experiment for the barometer sensor. The data clearly shows the decent and ascend with constant levels of corridor walking. Close inspection of the decent and ascent shows three steps in the sensor data, which corresponds to four single flights of stairs used to connect the different levels of the University building.

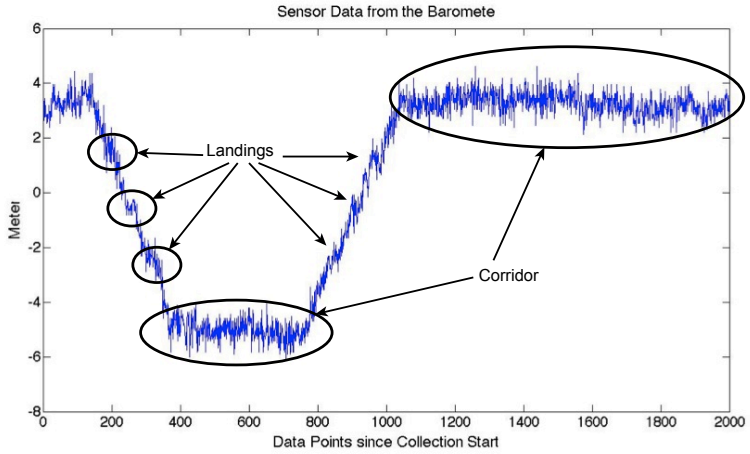


Figure 3–7: Altitude output from the barometer sensor (BMP085)

The on-board sensors allow not only the activity classification but can also be used for localization purposes or improved movement analysis. This technique is already used for quad-copters and recent developments in the area of MEMS sensors tries to combine 10 DoF on a single sensor [135] to push the technology into the area of indoor navigation where the use of the Global Positioning System (GPS) is limited, such as shopping malls and elderly care homes.

The experiment also highlighted the approximate communication range under clear line of sight conditions between the OHDAD and a smartphone to be 4 - 5 m. In addition, a current draw of approximately 75 mA with the BT wireless module attached but not

connected to an Android smartphone and around 55 mA while connected was measured. The drop in current is explained by a reduced power consumption of the BT module when not searching for a base station. A battery pack consisting of four rechargeable AA batteries (~2000 mAh) can power the OHDAD therefore for at least 26 hours (calculated using equation 3-1) even without a base station in range. Further tests showed a runtime of over 24 hours with fresh batteries.

$$Runtime_{OHDAD} = \frac{Source_{Charge}}{Consumption_{OHDAD}} \quad (3\text{-}1)$$

The last aspect investigated was the achievable sampling rate, which at 10 Hz is comparatively low for the relevant literature but an unavoidable limitation introduced by using the Arduino microcontroller. The impact of a low sampling frequency onto the classification accuracy is therefore a specific part of a parameter selection analysis (including sampling frequency, segmentation method, and window size) in Chapter 5. However, the outcome highlights that a low sampling frequency has only a minor to non-significant influence on the achievable classification accuracy.

3.3 Implementation of a New Software Simulation Environment

3.3.1 Motivation for a Simulation Environment

The OHDAD proposed above allows for long-term data collection and if worn over a long period will unquestionably be a very good source of data. The problem though, is the long lead-time required before meaningful data analysis can begin. Moreover, when worn under real conditions, an abnormal activity (such as a fall) might not occur during the collection process [136]. In addition, when working with test subjects outside the research group ethical consideration is required, which includes additional lead-time. Test subjects might even expect to be compensated, as stated in [89] whom paid two subject $15.00 each day during a study lasting 30 day.

Therefore a simulation environment with the aim to reduce the time required to produce long-term data, reduce the cost, and prevent ethical consideration was considered. A problem with this approach is that there does not exist a mathematical formula to model human activity acceleration. The simulation output should be based on real world data and not on arbitrarily selected sensor values. Therefore the simulation environment should use short activity segments that are collected by the OHDAD as a data input, in order to produce reasonable long-term human activity data.

The simulation environments found in the literature review in Chapter 2 did not reveal any on-body sensor simulation for the abnormality detection. The closest work to the expected long-term simulation environment was presented in [137]. Even though, their simulation environment is not openly available, the research paper presented two interesting arguments. The first one is that there exists no mathematical model for human behaviour simulation and the second that Markov chains can be used to model room occupancy. This approach is supported by [64], who derived probabilities for a Markov chain from real sensor data. Moreover, several monitoring systems use Markov Models for ADL classification [28], [55], [79], [138]. This led to the design of a simulation environment based on Markov chains and short-term activities collected by the OHDAD to decrease the time required for the long-term data generation. Therefore the following section introduces Markov chains and presents the design of the simulation environment, the selection of transition probabilities between activity states, and an evaluation approach, as there exist no mathematical model for a simulation validation.

3.3.2 Introduction to Markov Chains

The general definition of a Markov chain describes a time-discrete stochastic process, which transitions between the m states of a finite set $S = \{s_1, s_2 \dots s_m\}$. The transition between a state i and all possible j states is probabilistic with the requirement of $\sum_j P_{ij} = 1$. Furthermore, the transition probability P_{ij} has to be independent of past states so that the result of a random process X_n fulfils equation 3-2, with $j, i_0, \dots, i_{n-1}, i \in S$ and $n + 1 > n \dots > 0$ [139], [140].

$$P_{ij}(X_{n+1} = j | X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i) = P_{ij}(X_{n+1} = j | X_n = i) \quad (3\text{-}2)$$

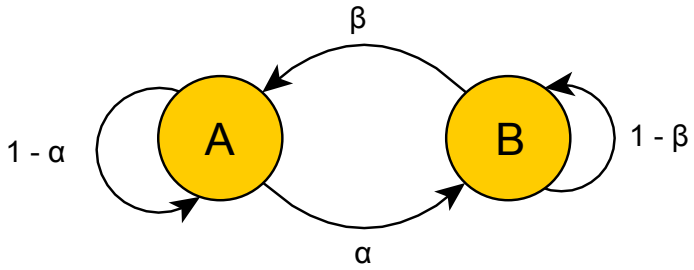


Figure 3–8: Two state Markov chain

Figure 3–8 illustrates a Markov chain for the finite set $S_n = \{A, B\}$. The different transition probabilities can be written as the transition matrix $P = \begin{bmatrix} P_{AA} & P_{AB} \\ P_{BA} & P_{BB} \end{bmatrix} =$

$\begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}$. Both rows sum up to one, hence fulfilling $\sum_j P_{Aj} = P_{AA} + P_{AB} = 1 - \alpha + \alpha = 1$ and $\sum_j P_{Bj} = P_{BA} + P_{BB} = \beta + 1 - \beta = 1$. When the start position S_0 of the Markov chain is known, for example A, it is possible to determine the probability for the system to be in state B after two time iterations ($n = 2$).

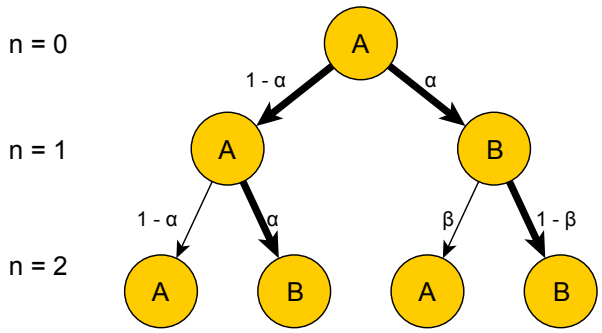


Figure 3–9: Possible state transition from A to B in two steps

Figure 3–9 illustrates the possible states for the system at times $n = \{0, 1, 2\}$. Using the initial state $S_0 = [1 \quad 0]$ and $P = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}$, it is possible to calculate the next state:

$$S_1 = S_0 * P = [1 \quad 0] * \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}$$

$$= [1 * (1-\alpha) + 0 * \beta \quad 1 * \alpha + 0 * (1-\beta)] = [1-\alpha \quad \alpha]$$

This represents the initial probability to move from state A to B. In the same way it is possible to calculate S_2. At $n = 2$, it is possible to move from state A to B or to stay in state B.

$$S_2 = S_1 * P = [1-\alpha \quad \alpha] * \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}$$

$$= [(1-\alpha) * (1-\alpha) + \alpha * \beta \quad (1-\alpha) * \alpha + \alpha * (1-\beta)]$$

$$= [(1-\alpha)^2 + \alpha * \beta \quad \alpha(2-\alpha-\beta)]$$

Based on this result, the probability to be in state B when the system started in state A after two steps is $\alpha(2-\alpha-\beta)$. Once α and β are determined the probabilities value can be calculated. For example, with $\alpha = 0.3$ and $\beta = 0.3$, $P_{AB}^2 = \alpha(2-\alpha-\beta) = 0.3 * (2-0.3-0.3) = 0.42$. For the work presented here, states of the set $S_n = \{s_1, s_2 \ldots s_m\}$ represent different ADL activities, described in Section 3.3.4, with the selection of the corresponding transition probability matrix P described in Section 3.3.4.1.

3.3.3 Overview of the Simulation Concept

The introduction to Markov chains presented above serves as the basis for the simulation concept illustrated in Figure 3–10. The simulation concept is built upon the following three tasks: 1) initial research investigation, 2) data collection with the OHDAD, and 3) configuration of the simulation environment. The latter requires the following four sub tasks: 1) Markov chain design, 2) selection of required time slots per day and days requiring a probability change, 3) produce a transition probability matrix P for each time slot and selected day change, and 4) selection of simulation runtime.

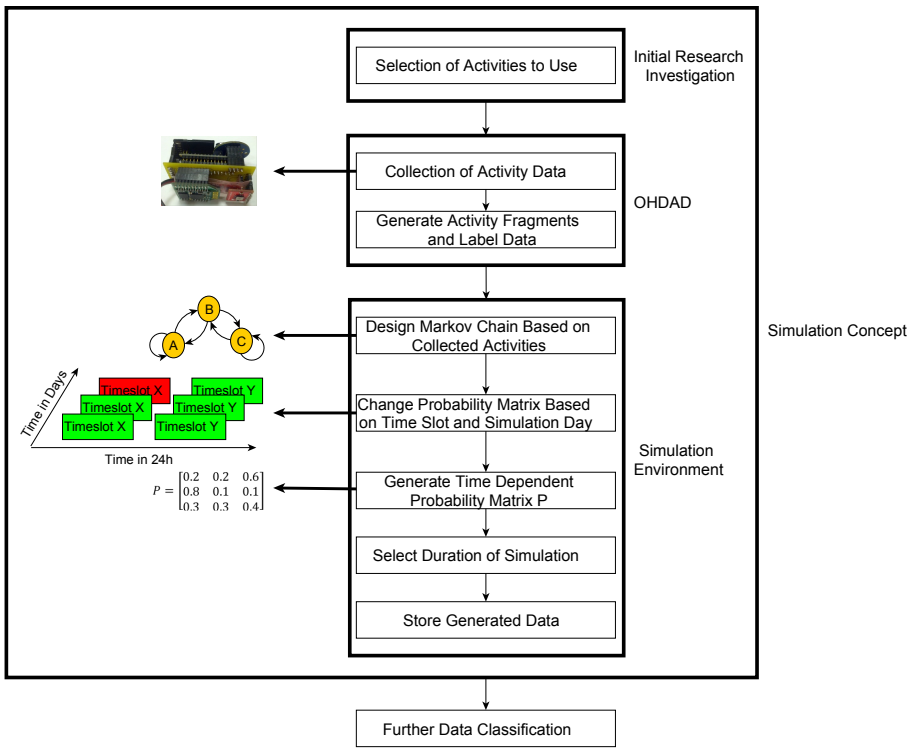


Figure 3–10: Necessary steps for the long-term data generation using OHDAD and the software simulation environment

The first task is the design of the experiment to determine the data required for the long-term data generation (see Section 3.3.4). The activity data is collected in the second stage using the OHDAD. The activity data can be recorded in a laboratory or non-laboratory environment in one continuous session. Activity extraction and labelling from the sensor log can be done post-collection using the recorded time stamps. The third task is the configuration of the simulation environment and should be based on the initial design thoughts from the first task. The first sub task is the design of the state transitions. While in principle, the individual states can be connected arbitrarily, good judgement should allow for meaningful connections. This process is explained in more detail in Section

3.3.4. The second sub task is the selection of time slots for each simulation day as activity probabilities can change in the process of a day. Another aspect is the change of activity probabilities based on the simulation runtime, for example to allow an abnormal activity only to occur during a certain period (day and time). The third task is the generation of transition probability matrix P defining the changes between each $activity_i$ to all possible $activity_j$. The matrix is dependent upon the selection of time slots and probability changes during the simulation runtime, as designed in the prior task. The selection of transition probabilities is further described in Section 3.3.4.1. The last task is to define the simulation runtime, which depends on the required length of the long-term data. After these parameters are specified the long-term data generation can begin. For each simulation day, collected and labelled short-term activity fragments are spliced based on the transition probability matrix P. Each simulation day is stored as a Matlab file for further processing using different monitoring approaches. The benefits of this simulation concept for the long-term data generation with short-term data fragments, is the relative simplicity to label ADLs instead of having test subjects record and label months of sensor data. Moreover, this approach also ensures that the collected data actually contains abnormality.

3.3.4 Activity Selection, Collection and Markov Chain Layout

The literature presented in Chapter 2 that was using Markov chains, concentrated on the generation of room occupancy patterns using Passive InfraRed (PIR) sensor data [64], [69], [137]. Though this work is focused on the long-term data generation of accelerometer sensor data from human activities, as it provides faster and direct information in case of an emergency. The simulation literature presented in Chapter 2 did not highlight the simulation of activities based on accelerometer sensor data. Furthermore, the simulated and collected PIR sensor information, found in the literature survey, did not allow the inferring of ADLs either. Therefore, only the activities highlighted in the short-term monitoring are considered for this simulation environment.

Table 3–4: Activities of interest found in the literature

Source	Activity
Ali, 2012 [77]	Step-Up, Step-Down, Sit-to-Stand
Lara, 2011 [78]	Walking, Running, Going Upstairs, Going Downstairs, Sitting
Zhu, 2010 [79], [138]	Sitting, Standing, Lying, Walking, Sit-to-Stand, Stand-to-Sit, Lie-to-Sit, Sit-to-Lie
Teixeira, 2009 [80]	Cooking, Eating, Brushing Teeth, Using Toilet Up/Down, Cleaning, Fetching Water, Taking out Trash
Patterson, 2005 [55]	Using the Bathroom, Making Oatmeal, Making Soft-boiled Eggs, Preparing Orange Juice, Making Coffee, Making Tea, Making or Answering a Phone Call, Taking out the Trash, Setting the Table, Eating Breakfast, Clearing the Table
Jantaraprim, 2010 [81]	Forward Fall, Backward Fall, Left and Right Side Fall, Sit-to-Stand, Stand-To-Sit, Sit-to-Lie, Lie-To-Sit, bend down to pick up an object, Walk for 2m
Nguyen, 2009 [82]	Fall, Sitting, Standing, Lying, Walking (Slow and Fast), Going Upstairs, Going Downstairs, Sit-to-Stand, Stand-to-Sit, Stand-to-Lying, Lying-to-Stand
Dinh, 2009 [83]	Four times of fall: collapse, forward, backward, sideward
Nocua, 2009 [141]	Standing-to-Lying, Fall

Table 3–4 presents the publications and the collected ADLs. Common high-level activities are: walking, standing, sitting, lying and their transitions stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit. These activities are therefore selected as normal states for the simulation environment, with abnormal data represented by a forward fall. The fine-grained activities, such as making coffee or tea, could also be used but their usefulness for the abnormality detection is not given, as the OHDAD data is not suitable for this kind of differentiation.

In [142], it is pointed out that a data mining algorithm requires training and test sets in order to be operational. On one side, literature concerning Artificial Immune Systems (AIS) highlighted that the algorithm does not need a specific training set, as it learns continuously from on-going data. On the other side, supervised classification algorithms that are compared to AIS require this information. The works presented in Table 3–4 used repetitions in the range of 1 to 20, with some not stating this information at all. The selection of repetitions is influenced by two factors. On one side, more repetitions allow for a larger training set. On the other side, the test subject is required to repeat the same activity more often. This process does not only increase the necessary time for the data collection, it also increases the possibility of an injury of the subject during dangerous activities, such as a fall. Since AAL environments are specifically aimed at elderly, who have a weaker body structure, activity repetitions for the training set should be kept low. Therefore, the author oriented the number of repetitions used in this work at the lower level found in the literature. In the authors viewpoint, five recorded activity repetitions provide enough variation in the different activities, without keeping the test subject too long in the laboratory and expose him to too much danger of hurting himself.

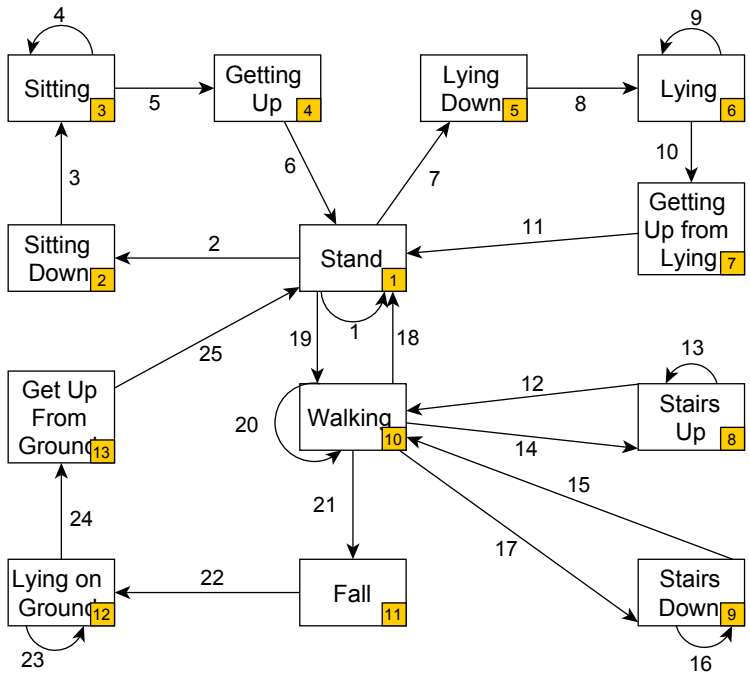


Figure 3–11: Markov chain layout defined for the simulation environment

Several different Markov chain layouts are possible with the collected ADLs. The selection of transition probabilities will be covered in the next section. Even though, the literature does not elaborate on how activities should be modelled based on Markov chains, [143] suggests to use one's own expert knowledge, when no data is available. Figure 3–11 illustrates the connection of thirteen individual Markov states based on four activities and their corresponding transition activities and the abnormality. It is determined that the activities going up/down stairs and the forward fall have to follow the walking state. Other abnormal activities, such as falling the stairs up/down, or sliding of the chair could be added to the Markov chain layout but were outside the work here presented. Especially as fall on the stairs can lead to serious injuries, it should only be recorded using sufficient padding on the test subject or a specially trained person, such as a stunt double. The activities, representing the stairs movement, can directly merge into the walking movement, while a fall, as it contains an undefined length of lying on the ground, should be followed by standing. This would emulate the process of catching one's breath or being dizzy after falling. The two other main activities, sitting and lying, require furniture to do so, therefore a test subject cannot directly move into this activity from walking. The standing fragment allows for a smoother transition, emulating a stop in front of the furniture (chair, bed, etc.). One of the limitations of this layout is the constant use of the same activity fragment. Here a trade off between size of training and

test set is required. It is, however, the author's belief that with this method, a normal day can be sufficiently simulated.

3.3.4.1 Selection of Transition Probability

The last aspect that needs to be decided upon is the selection of transition probabilities between the single Markov states. While the work of [64] influenced the general design of the simulation environment, their method of using past sensor data to define state transition probabilities cannot be used, as no data is available. Instead, in [63] the probability to move from state i to j after one time instant k is set to 50% ($P_{ij}^{k} = 0.5$), as illustrated in in Figure 3–12.

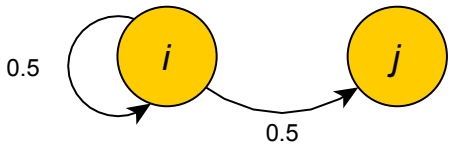


Figure 3–12: Stateprobability of 50%

From there it follows that to stay in the same state i after n time steps, the probability has to be $P_{ij}^{k} * P_{ij}^{k} * \ldots * P_{ij}^{k} = (P_{ij}^{k})^{n} = 0.5$, which translates to $P_{ij}^{k} = (0.5)^{1/n}$. Based on this assumption, probability values are chosen to simulate discrete PIR sensor events within a minute data resolution. In the beginning the selected transition probabilities represent a stable condition that can be later changed to introduce e.g. nocturnal urination problem through longer stays in the bathroom during the night. Although good results are reported, it is pointed out that it is nearly impossible to perfectly simulate a human activity. It should merely be a tool to provide adequate data to test monitoring systems. This approach, while not specifically stated, follows the in [143] given recommendation of being a self-expert, meaning to use one owns expertise in the field.

The author decided to follow the above approach even though the simulation environment uses the length of used activity fragments to determine the simulation time instead of discrete one minute steps as [63]. Therefore, the transition probability calculation needs to be adjusted and based on the repetition of an activity. The repetition can then be used in conjunction with the activity length to determine the approximate time the system will spend in this state. The time spend executing an activity can be an interesting aspect of detecting abnormality, while not part of the work presented here, spending a considerable amount of time in one activity should lead to an abnormality warning. Knowing that the five main activities are short 10 s recordings, it is possible to

define a minute of activity using $n = \frac{y}{6}$, with y being the number of repetitions. The advantage to use an equal probability of 50% to stay in the same state, as described in [63], is the possibility to define the average time spend in the individual state. Other probability combinations would result in an imbalance between the states and an average time could not be defined. This would also require the individual calculation for each state and increase the complexity of the proposed simulation. Therefore the 50% probability to stay in the same state is adopted, resulting in the modified formula for $P_{ij}^{k} = (0.5)^{\frac{1}{n*6}} = (0.5)^{\frac{1}{y}}$.

Table 3–5: Conversion table to determine Markov transition probabilities for a fixed set of activity times in minutes (Adapted from: [63])

Activity expected to last n minutes	Activity expected to be repeated y times	Probability Pij(k) required to continiue with activity I for n minutes	Probability of continiueing with activity i after y repetitions
0.5	3	0.7937	1/2
1	6	0.8909	
2	12	0.9439	
3	18	0.9622	
10	60	0.9885	
30	180	0.9962	
60	360	0.9981	
120	720	0.9990	
180	1080	0.9994	

With the correlation between minutes of activity and number of repetitions defined, Table 3–5 is generated for a general recommendation when preparing the transition probability matrix. The table presents probabilities for short activity times, as the ADLs are supposed to take place in a domestic environment. For example, going up or down stairs will only be a short event in a two story house and the assumption that test subjects do not have problems with this kind of activity. Moreover, elderly might even move houses just to avoid stairs all together (barrier free living). Another short activity will be walking. Even though it might be frequent, prolonged times of walking will be a rare event and most likely outside the domestic environment. Activities lasting around 30 – 60 minutes can happen when standing in the kitchen preparing food or when pursuing comfortable activities, such as sitting and napping. The two longer periods of 120 and 180 minutes are mostly interesting during the night to simulate long periods of sleep.

3.3.4.2 Definition of Transition Probability Matrix

The use of only one probability table during the day would result in a static activity distribution and while legitimate for an abnormal activity detection system that does not monitor the start time of an activity, it is not close to actual human behaviour. For example in [63] and [144], the authors highlighted seven unevenly spaced timeslot throughout the day (awakening: 7:30 – 8:30, morning: 8:31 – 12:00, lunch time: 12:01 – 13:30, afternoon: 13:31 – 19:00, evening: 19:01 – 21:00, bedtime: 21:01 – 23:59, night: 00:00 – 7:20). Furthermore, in [145], a circadian activity rhythm (CAR) was introduced to represent a cyclic behaviour of a human during a day. Therefore, the simulation environment developed as part of this work, allows a 24h day to be divided into X equally spaced timeslots.

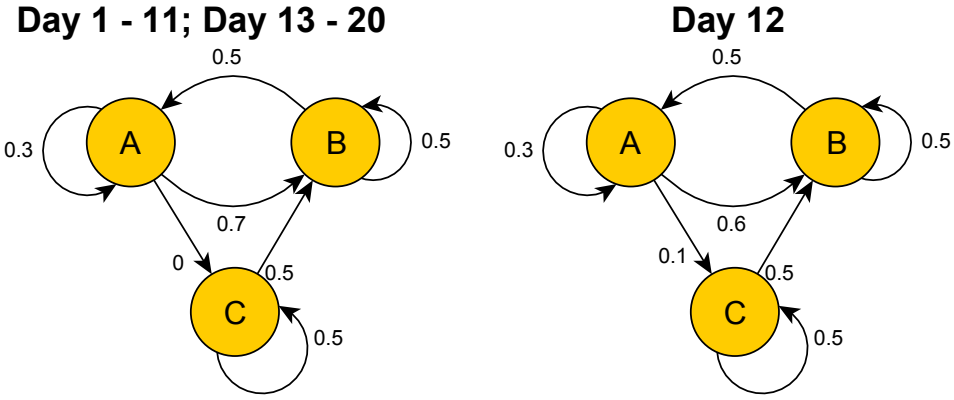


Figure 3–13: Changes in state transitions probabilities in a Markov chain

Another aspect of transition probabilities is the ability to modify this parameter depending on the simulation day, for example to generate abnormal activities at specific times or simulate weekdays and weekends. This method also allows a pre-defined training time before abnormalities are introduced. This behaviour can be modelled as illustrated in Figure 3–13 using Markov chains, where state A and B represent normal activities and C an abnormal activity. During the simulation day 1 -11 and 13 - 20 it is not possible to get from state A to C, hence no abnormality will be introduced into the long-term data. During day 12 the same layout is presented, with a probability of 0.1 to move from state A to the abnormality state C. At the same time the probability of going from A to B is reduced by 0.1 to keep $\sum_j^n P_{ij} = 1$ fulfilled.

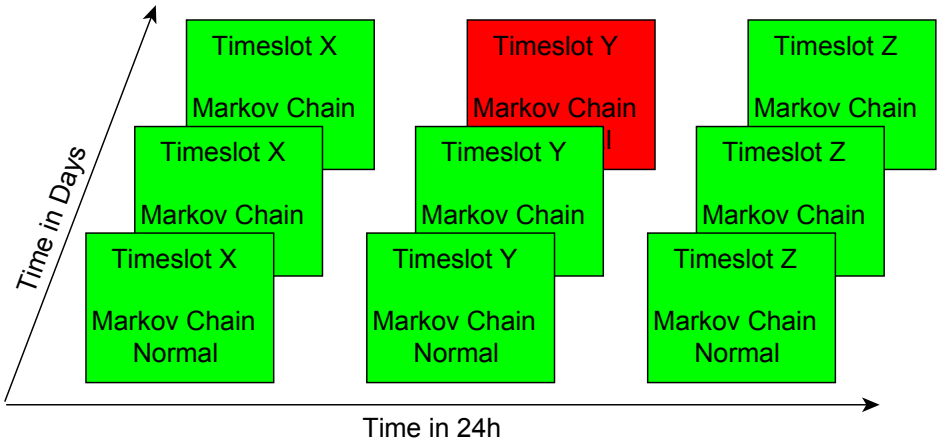


Figure 3–14: Distribution of transition probability matrixes for the long-term data generation

This approach leads to several transition probability matrixes P, as illustrated in Figure 3–14. The result is a $m\ x\ n$ matrix, with m being simulation days and n representing the different timeslots:

$$TransitionMatrix = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{m1} & \dots & P_{mn} \end{bmatrix}$$

In the illustration, the simulation is only three days long and each day is divided into three 8 hours blocks ($\frac{24\ hour}{3} = 8\ hours$), requiring nine independent transition probability matrixes *P.* The task of generating these matrixes can be time consuming and repetitive, especially when consecutive days are using the same transition probabilities for the same time slots. Therefore, to simplify this process, the software simulation only uses the next row in the $TransitionMatrix$ when specifically instructed to. For the in Figure 3–14 illustrated example, only six probability matrixes are required with the specific transition change defined at day three.

The approach presented above allows for better natural human behaviour and better flexibility during the simulation run time. The use of Markov chains enables the design of versatile test scenarios for the monitoring systems, which can be fully controlled in the means of transition probabilities and state changes. This approach can provide vital information, concerning the monitoring system or data requirements, during the possible month long lead-time of the data collection with a real test subject. The software simulation environment represents a major improvement on cases where the collected long-term data requires further time for the labelling of activities. In [134], the annotation ratio was described with a staggering 7 hours to every 30 min of data collection under laboratory conditions.

3.3.5 Evaluation Approach of the Simulation Environment

3.3.5.1 Validation of Markov Chain

After the concept and design of the simulation environment is introduced, the correct simulation behaviour needs to be validated. This process would require either a mathematical model or past sensor data to derive a model for comparison. It is therefore not possible to validate the simulation output in the sense of [66] and [65], as both determine the output variation between a real and the programmed model. Instead, the approach of an evaluation needs to be considered, which after [67] might be the best approach for a complex system, such as human behaviour. The proposed evaluation process consists of the visual comparison between Passive InfraRed (PIR) and high-level activity label graphs presented in the works of Virone et al. [69], [70] [63]. If with the simulation environment presented here, similar looking graphs can be generated, at least the correct working of the simulation can be assumed and hence providing at least the knowledge that later chosen transition probabilities will lead to the expected simulation output.

Figure 3–15.a presents a graph from [70], which was generated using the simulation environment presented in [63]. While the authors' earlier work used room occupancy as the data label, the new data is labelled using passive activity data. Moreover, the activity data is only a label and not based on any pre-recorded accelerometer data, as presented in this work. Figure 3–15.b presents a graph from [69], which highlights room occupancy in minutes for the weekends of a real test subject (subject 2). Both figures present normal behaviour of the test subjects.

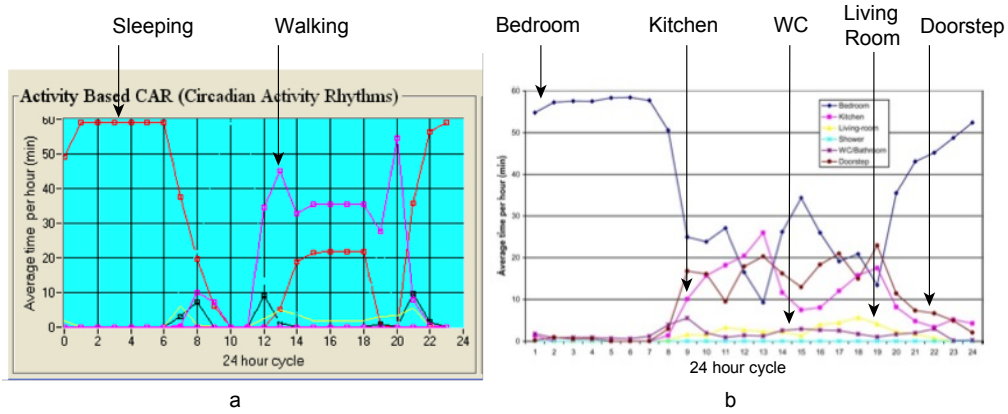


Figure 3–15. Circadian Activity Rhythms representation (Source: [69], [70])

For Figure 3–15.a, the first observation is, sleeping is a regular activity from 21 – 9 o'clock and during midday (14 – 18 o'clock). The second observation is the long period of

walking between 12 – 20 o'clock. The other activities were not further labelled. For Figure 3–15.b, it can be observed that the bedroom occupancy correlates to the sleeping activity in Figure 3–15.a. Furthermore, the PIR data shows that the test subject spends more time in the kitchen than in the living room. Both figures align with [6], [69] who suggest that under normal circumstances, a human will be more active during the day.

Table 3–6: Transition probabilities used for the software simulation evaluation

Transition		Transition Number	Timeslot			
From	To		1	2	3	4
			0 - 6	6 - 12	12 - 18	18 - 24
Stand		1	0.794	0.9772	0.8909	0.9772
Stand	Sitting Down	2	0.006	0.01	0.06	0.01
Sitting Down	Sitting	3	1			
Sitting		4	0.962	0.9772	0.9962	0.9962
Sitting	Getting Up	5	0.038	0.0228	0.0038	0.0038
Getting Up	Stand	6	1			
Stand	Lying Down	7	0.15	0.0028	0.0091	0.0028
Lying Down	Lying	8	1			
Lying		9	0.999	0.9772	0.962	0.998
Lying	Getting Up from Lying	10	0.001	0.0228	0.038	0.002
Getting Up from Lying	Stand	11	1			
Stairs Up	Walking	12	0.206			
Stairs Up		13	0.794			
Walking	Stairs Up	14	0.008	0.0064	0.0064	0.006
Walking	Stairs Down	15	0.008	0.0064	0.0064	0.006
Stairs Down		16	0.794			
Stairs Down	Walking	17	0.206			
Walking	Stand	18	0.04	0.025	0.025	0.044
Stand	Walking	19	0.05	0.01	0.04	0.01
Walking		20	0.944	0.9622	0.9622	0.944
Walking	Fall	21	0			
Fall	Lying on Ground	22	1			
Lying on Ground		23	0.944			
Lying on Ground	Get Up from Ground	24	0.056			
Get Up from Ground	Stand	25	1			

Based on the information presented in Figure 3–15.a and Figure 3–15.b a general behaviour model had to be derived to decide on the required transition probabilities. Table 3–6 presents the 25 different transition probabilities for the Markov chain layout introduced in Figure 3–11 in Section 3.3.4. The transition from / to column corresponds to the thirteen different Markov states and the 25 transition numbers to the different state transitions between these states as defined by the Markov chain. For the evaluation process, the day is divided into four equally spaced timeslots. The transition probabilities can be independently adjusted for the night period (0 – 6 o'clock, timeslot 1), morning (6 – 12 o'clock, timeslot 2), midday (12 – 18, timeslot 3), and evening (18 – 24 o'clock, timeslot 4). For each timeslot the average time spent in one of the four main ADLs (stand, sitting, lying, and walking) need to be defined and then converted to the transition probability P_{ii}, introduced in Section 3.3.4.2, using Table 3–5. The transition probabilities P_{ij} to the j adjacent states have to be determined so that $\sum_j P_{ij} = 1 - P_{ii}$. In addition, transition activities cannot directly repeat themselves as they define a physical change in the test subjects position, such as 'sitting down' changes from standing to

sitting and would require the activity 'getting up' before a test subject could repeat the sitting down motion. Therefore, these activities have a $P_{ii} = 0$ and the only adjacent state transition a probability of $\sum_{j=1}^{1} P_{ij} = P_{ij} = 1 - P_{ii} = 1 - 0 = 1$. Hence, the transition probability of 1 for the transition numbers 3, 6, 8, 11, 22, and 25 in Table 3–6. In addition, this evaluation does not include abnormal activity, and therefore a transition probability of 0 for the only transition to the 'fall' state (transition 21), as defined in Figure 3–11 in Section 3.3.4.

For timeslot 1, it was determined that the main activity should be sleeping, therefore the average time n was set to 120 min, or a transition probability of 0.999 see also transition number 9 in Table 3–6. The average time for standing is 0.5 min (transition probability of 0.794), sitting is determined as 3 min (transition probability of 0.962) for the toilette use, and walking is set for 2 min (transition probability of 0.944). During timeslot 2, the average time for sleeping is decreased to only 10 min (transition probability of 0.977). Furthermore, the time of standing and sitting are increased to 10 min, too. This should represent an increased activity of the test subject, for example in the kitchen preparing and eating food. The average walking time is only slightly increased to 3 min (probability of 0.9622), as 3 minutes are still a significantly long time in a domestic environment. This is also where the author disagrees with the in [70] presented simulation output (see Figure 3–15.a). The presented 6 hours of walking during an 8 hour time period is also not supported by the graph presented in Figure 3–15.b from the same authors [69]. Timeslot 3 reduces lying further down to 3 minutes (probability of 0.9622) and standing to 1 min (probability of 0.8909). Walking keeps the average of 3 minutes, but the time the test subject sits is increased to 30 min (probability of 0.9962). This would model watching TV or reading a book through midday. For the last timeslot the average time of standing is set to 10 min (probability of 0.9772), sitting stays at 30 min, lying is again increased to 60 min (probability of 0.998), and walking slightly decreased to 2 min (0.944). The increase in standing emulates the preparation for dinner and getting ready for bed, hence the increase in the lying activity as well. With P_{ii} defined, the P_{ij} transition probabilities were arbitrarily divided.

Based on these parameters, the simulation environment produced 10 normal days. To be able to compare the output of the simulation environment presented here with the in Figure 3–15.a and Figure 3–15.b illustrated graphs, the individual activities are evaluated on an hourly bases. Figure 3–16 illustrates the average time spend in each

activity over the 10 simulation days. The graph shows similarities to the anticipated behaviour. Timeslot 1 is dominated by a sleeping cycle, while timeslot 3 by the sitting activity. A slightly unexpected behaviour can be seen in timeslot 2 and 4. For the former the standing activity might be too dominant, while the latter reduces the time the test person spends in the sleeping activity. If a more natural representation of activities during the day is required, a finer grained time spectrum that divides the day in more than four timeslots might be necessary. This would allow a more sensitive approach, with more intermittent probability steps. However, a general similarity can be seen between the activity output by the simulation environment developed for this work here and the PIR representations presented in Figure 3–15.a and Figure 3–15.b by Virone et al. Furthermore, the different approaches used in this work, are purely based on sensor data that is not influenced by the probability distribution of activities throughout the day.

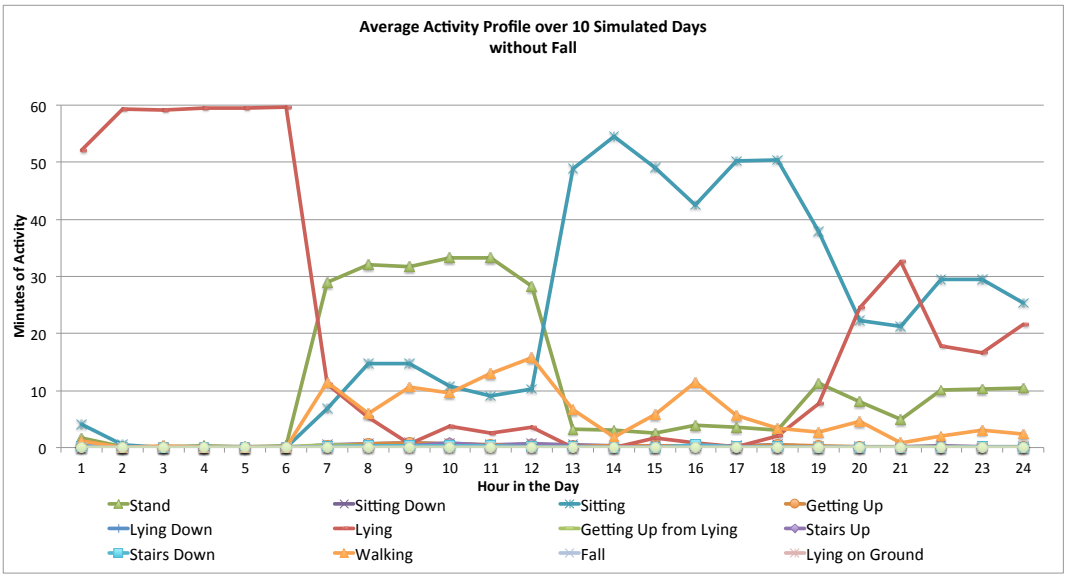


Figure 3–16: Distribution of simulated activites, summarised over 10 days

3.3.5.2 Comparison of Continuously Collected Data and Spliced Data

Another approach of the simulation evaluation is the visual comparison between continuously collected sensor data by the OHDAD and the output of spliced short-term data fragments generated using the simulation environment at the data level. Figure 3–17 and Figure 3–18 illustrate the X-, Y-, and Z-axis of the tri-axial accelerometer sensor, for a fall directly captured with the OHDAD platform and generated by the simulation environment, respectively. Even though, the representation of the simulation output includes several smaller walking fragments prior to the actual fall, the visual inspection

does not highlight any apparent separation in the data output. In addition, both figures show similar acceleration values prior to the fall, with the exception that the simulation environment used a data fragment, where the test subject stood still for several seconds before a fall occurred. Furthermore, both figures show a high acceleration at the occurrence of the fall. The comparison presented here provides confidence in using spliced short-term sensor data for the generation of long-term data. The splicing method of sensor data was also presented by the author of this thesis in [97].

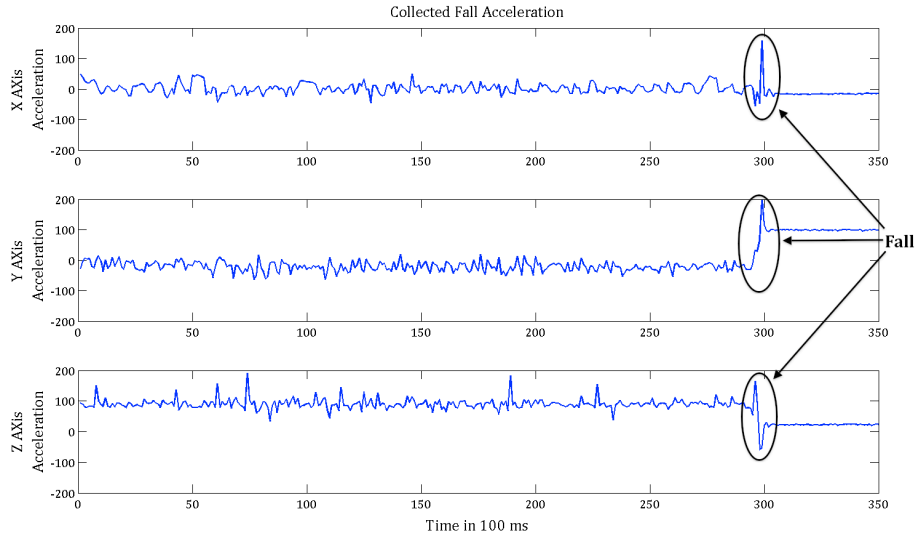


Figure 3–17: Three-axis accelerometer data of a fall event recorded by the OHDAD

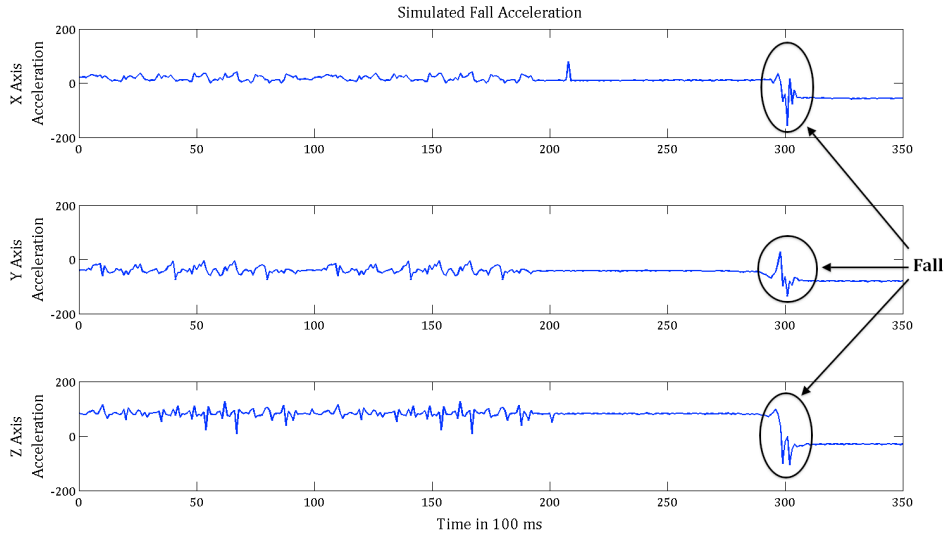


Figure 3–18: Three-axis accelerometer data of a fall event generated using the software simulation environment

3.4 Summary

This Chapter has described the OHDAD and accompanying long-term data simulation environment developed as part of the work presented in this thesis. The OHDAD has

enabled the author to collect short- as well as long-term 10 DoF data of human activity. All necessary files required to reproduce the OHDAD have been publicly published in [131]. In combination with the long-term data simulator, the author can design simple as well as complex home monitoring scenarios of various time lengths. The simulation environment is able to generate long-term activity data using collected sensor data from the OHDAD. The simulation supports an unlimited amount of activities and flexible adjustments to the Markov transition probability matrixes. The human behaviour simulated using a Markov chain, was validated against real data, by comparing the simulated activity distribution with PIR sensor data presented by Virone et al. While the focus of the works was different (Activities and PIR), there is good agreement between simulation results, which provides confidence in using the data generated for experimentation. Furthermore, the author's approach to the detection of abnormal activity enables a faster reaction time in the event of an abnormality, as acceleration data allows for the instant data analysis compared to the hourly averaged room occupancy.

4 Artificial Immune System-Based Monitoring System: Design and Implementation

4.1 Introduction

This chapter aims to introduce the concept and explain the motivation behind the use of an Artificial Immune System (AIS), which belongs to the Computational Immunology (CI) in the Artificial Intelligence (AI) field, as a novel approach for the detection of abnormal activity in Ambient Assisted Living (AAL) applications.

General AIS design concepts are presented and a novel detector seeding method is introduced and discussed in detail. Afterwards, the developed monitoring system based on the introduced AIS design is presented and described.

4.2 Motivation For The Use Of AIS in AAL Applications

The literature search has highlighted different approaches to AAL monitoring, with the common idea to understand what "normal" behaviour means in the context of AAL. While it is interesting to characterise normality, several works [85], [86], [93], [145], [146] have highlighted the usefulness and necessity of detecting abnormality or deviation from normal behaviour, as these events can be an indicator for current or soon to happen health problems [147]. In [87], it was highlighted that abnormality detection can be addressed using the following two different methods: profiling and discriminating. The latter requires normal as well as abnormal data to be present for the learning algorithms. As stated in [78], gathering this abnormal data might result in an adaptation barrier, as performing certain activities is too dangerous for a test subject. Even when considering "stunt doubles" performing dangerous activities [148], the slight differences to the actual test subject introduce uncertainty [78]. Another source of uncertainty is the fuzzy definition of an abnormality event itself [147]. In [87] the authors state that countless forms of abnormality are imaginable in the context of AAL, which makes defining and collecting every abnormality impossible. In [106] it was highlighted that the problem with unknown abnormalities is the assumption of normality until a known abnormality occurs. Even though their application area was hardware fault testing, a similar problem can be expected in the AAL context. Therefore, Elbert's et al. profiling method mentioned above is of interest, as it only requires normal

data to be present in the data set and expects deviation to have its source in abnormality. Furthermore, the profiling method is divided into a uniform and individual approach. The author of this thesis believes that abnormal activity needs to be detected at an individual level, as humans can have their own lifestyle, which will change over the years. In AAL, anomaly detection has received only minor attention [147], therefore the following sections will outline the development of an anomaly detection system based on AIS, which to the author's best knowledge has not been used in this application domain in AAL before.

4.3 Artificial Immune System

4.3.1 Background

The initial idea to use an immunology inspired algorithm for the detection of abnormality was presented by Dasgupta et al. [149]. Their algorithm was inspired by the negative selection mechanism of the immune system, as it allows for the differentiation between a self- and non-self set. The self-set represents normal conditions and the non-self abnormality, or the deviation from the normal condition by a given margin. From this initial design for abnormality detection, other implementation approaches emerged to increase the application field of AIS. The four main approaches highlighted in the literature [150], [151] are presented in Table 4–1.

Table 4–1: Implementation approaches for the Artificial Immune System

Implementation Approach	Typical Application Field
Clonal Selection	Optimisation, Classification
Immune Networks	Clustering, Data Visualization
Danger Based	Anomaly Detection
Negative Selection	Anomaly Detection

For the work presented here Clonal Selection and Immune Networks are not of interest, as they are concerned with classification and clustering of data. Therefore, the work is limited to Danger Theory (DT) and Negative Selection (NS); with the latter being the main implementation approach in Computer Immunology (CI) algorithms, which model particular cases of AIS [150].

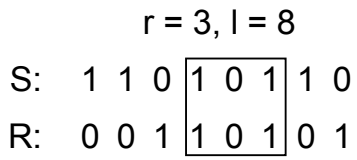


Figure 4–1: R-contiguous matching rule as presented by [152]

Based on [152] the principle of NS is:

- *Self* is determined by a given string set S with a defined string length of l and a finite alphabet A, for example $A = \{0, 1\}$.
- A detector set R containing x elements, defined over the same finite alphabet A and the length l, is randomly generated.
- All x elements of set R are matched to each string in the self-set S. Only detectors that match fewer than r contiguous positions are kept, with r being a parameter to be determined. Figure 4–1 illustrates a match of three continuous segments of set S and R, with $l = 8$. If r is defined as $r \leq 3$, the illustrated detector would be removed from the detector set R because too many positions are matching part of the self-set S. In the traditional sense of AI, S can be compared to a training set for the detector set R.
- New data S^* is matched with each surviving detector in set R to determine if it is part of the *self-* or *non-self* set. A match in at least r contiguous positions suggests the data to be part of the *non-self* set and results in an alarm condition. S^* can be compared to the test set in AI.

In [151], DT is described as having its theoretical background in immunology. The theory states that just because data in S^* activates a detector, it is not automatically an anomaly, as set S might be incomplete. On the immunology level of the Human Immune System (HIS), not every abnormality automatically results in an immune response. Antigen Presenting Cells (APCs) are activated by an additional signal that is only transmitted by injured cells and not by healthy cells. The transmission of the danger signal will create a danger zone and only antibodies in this zone can be stimulated. For the NS algorithm, this would mean an abnormality could only result in an alarm, if a second event is triggered simultaneously (co-stimulation). The problem with this approach is that the exact working principles of the danger signal are not yet fully understood. In [153] low or high memory usage, inappropriate disk activity, unexpected frequency of file changes or missing normal signals are highlighted as possible sources of a danger signal in the area of computer security. As there is no directly visible improvement for the danger signal for the application in AAL, it is the author's firm belief that for now it is sufficient to develop the monitoring algorithm purely based on the NS algorithm.

4.3.2 Framework

The NS algorithm described above has two limitations for AAL applications. Firstly, it requires a self-set *S* before the generation of a detector set *R*. This would make a-priori knowledge of a test subject necessary. Secondly, a normal (e.g. ageing related) change in the self-set *S* may result in new normal data points in data set *S** outside the normal margin (see Section 4.3.1 above). This would result in constant matched detectors of set *R*. The resulting alarms, while correct at the time of the initial learning period, would now be all false alarms.

In [154] and [98], the authors introduce a general framework for ARTificial Immune System (ARTIS). This framework improves on the two limitations, requiring a-priori knowledge and not being adaptable to changes in the self-set. For the modifications to work, the new framework requires abnormality to be an infrequent event, which in the context of AAL is a reasonable assumption to make, as subjects with frequent abnormalities might require a more personal care assistance. The modified framework allows for the use of AIS in combination with NS to be the foundation of the work described in this thesis.

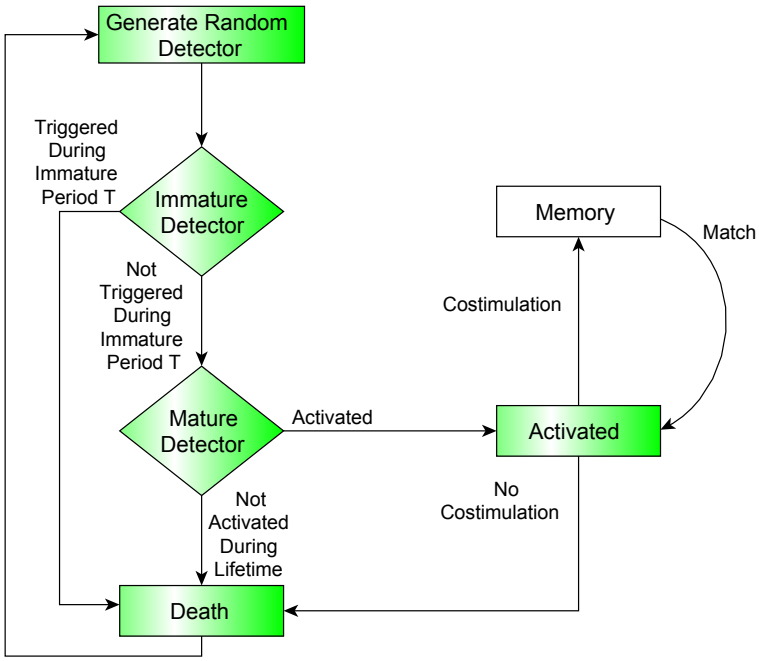


Figure 4–2: AIS detector life stages (Source: [98])

Hofmeyr et al. extended on the following aspects of AIS:

- Training the detector: A complete self-set *S* is not required when generating detectors.

- Memory usage: An activated mature detector is stored separately for a faster response time during later runtime.
- Sensitivity: An activation threshold is dynamically adjusted (decreased / increased) based on the number of currently activated detectors.
- Co-stimulation: An alarm condition has to be verified by a human.
- Lifecycle of a detector: A generated detector has a given lifespan to prevent a static system and be able to adapt to changes in the self-set S.
- Representation: A binary permutation mask is applied to generate different detector sets.
- Response: While not implemented, it was hypothesised that different types of responses are required for different anomalies.

For this work, the training of detectors (aspect 1) and the life cycle of a detector (aspect 5) are of interest. Figure 4–2 illustrates the new lifecycle of a detector. From the presented framework, only the green segments (Generate Random Detector, Immature Detector, Mature Detector, Activated, and Death) are of interest for this work. Randomly generated detectors start in an immature phase in the set $R_{Immature}$. Over a period of T these detectors will be matched using r-contiguous positions with elements from data set S^*, which can contain normal as well as abnormal data points. If the immature detector is not activated during period T it moves into the mature detector set R, in the original sense of NS, while an activation during period T results in the removal from set $R_{Immature}$ to be replaced by a new random immature detector. When a mature detector matches data from set S^* at r-contiguous positions, it does not directly result in an alarm. At first a detector moves into the activation stage, which requires additional stimulus (co-stimulation) to result in an alarm. Hofmeyr et al. implemented the co-stimulation in their work with a human operator who is notified and given a certain amount of time to react to the signal whenever a possible security threat is detected. If the operator reacts to the signal in time, the activated detector becomes a memory detector, otherwise a new immature detector will replace it. Memory detectors are similar to mature detectors, with the difference that they are normally more specific (increased r value for the r-contiguous matching) and do not have a specific lifespan. Hofmeyr et al. mentioned a resource limited memory array, which would require a competition between old and new memory detectors once all the memory positions are filled. While the aspect of competitions for memory positions is interesting, it is outside the scope to cover them

completely in this work. This work presented here concentrates on detector generation and the three detector stages: immature, mature, and activated.

4.3.3 Data Presentation

The initial NS algorithm stated that the self-set and the randomly generated detectors should be from a finite alphabet A. From the background of AIS in computer science A normally belongs to the set of $A = \{0, 1\}$. This resulted in several binary representations [98], [150], [155], [156]. The first publication from the author proposing the idea of AIS for the detection of abnormal activity suggested following the same direction [97]. It was anticipated to use the four following binary attributes: X-Axis acceleration, Y-Axis acceleration, Z-Axis acceleration, and acceleration average (calculated over the last 5 data samples). This would have resulted in 32-Bit binary variables, which could have used either r-contiguous or the Hamming distance for the matching rule. The two problems with this approach, were longer computational time, as collected accelerometer data had to be converted into binary values as well as complicated visualization. In [157] a real valued search space is proposed, which simplifies the visualization and removes the conversion step and is therefore preferred. While in principle, AIS is an n-dimensional approach, literature suggests to keep the possible search space dimension below five [158]. The author has therefore limited his approach to three dimensions, represented by the X, Y and Z acceleration axes of the OHDAD presented in Chapter 3.

4.3.4 Representation of the Self-Set

An element of the self-set S is in its original expression just an individual data point. [159] proposes that this does not need to be the case. He argues, that the existence of a single data point (be it normal or abnormal) in the search space increases the probability of additional data points of the same class in the surrounding area. The authors suggested a self-radius to allow for a generalization around each element of self-set S. A self-radius of zero (no generalization) is called aggressive (see Figure 4–3.a), while the existence of a radius $(\phi) > 0$ is called conservative (see Figure 4–3.b). It was highlighted that a large radius allows covering the self-set S with fewer self-elements. The disadvantage though, is the increased probability of missing abnormal data points, which are in close proximity to the self-set, as parts of the non-self set is covered by the normal self-radius. The data presentation pursued is dependent on the application field. The

advantage of presenting the self-set with less elements is ineffective, as the work presented here has no a-priori known self-set and a continuous stream of new data will be presented. Moreover, false alarms are preferred over missed alarms, hence the aggressive presentation is pursued.

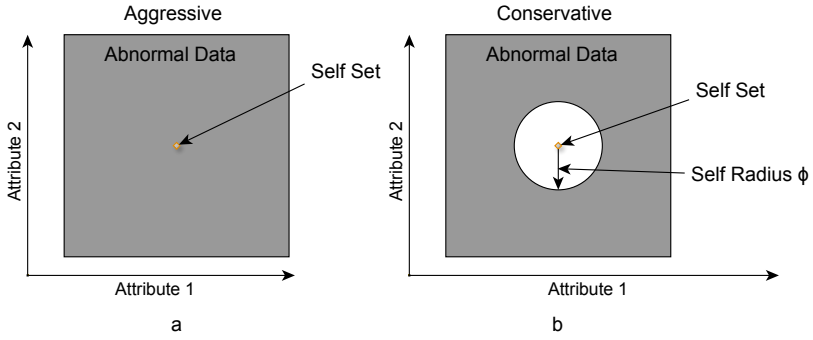


Figure 4–3: Presentation of the self-radius around a self-set as presented in [159]

4.3.5 Artificial Immune System Detector

4.3.5.1 Introduction

After the self-set S is defined, the elements of the detector set R need to be investigated. Detectors can be seen as the main component of the AIS, as they are the alarm component of the algorithm. Each detector has three parameters.

1. Position in the search space (Detector Centre Point (DCP))
2. Detection affinity area (activation area)
3. Detector age, in regards to immature and mature phase

Figure 4–4 illustrates the concept of the DCP in the search space, which is a general representation and not limited to accelerometer data that will be used in this work. An affinity area is defined around each DCP that acts as the detector's activation area. If data input falls into this radius the detector will be activated. Literature highlighted the use of Manhattan and Euclidean distance as possible methods to calculate the distance between the detector centre point and a data point, with the Euclidean distance (see equation 4-1) used the most [150], [160]. The p parameter used in equation 4-1 represents the n-dimensional position vector for the $DCP = (Attribute_1, .., Attribute_n)$, while q stands for the n-dimensional $DataPoint = (Attribute_1, .., Attribute_n)$, with $n \leq 3$ for the work presented here.

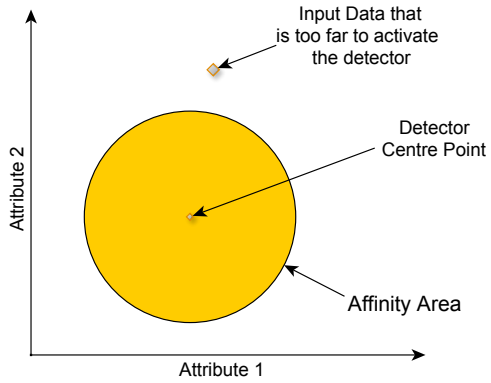


Figure 4–4: Visualisation of an affinity area around a single detector

$$Distance(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad (4\text{-}1)$$

4.3.5.2 Lifespan

The main advantage of the framework introduced in [98] is that it allows the self-set *S* to be presented over time without the need to store any past data. In this framework, a newly generated detector starts in an immature phase, which compares to a learning phase and prevents detectors from raising false alarms. This stage determines if a detector covers the self-set or the non-self set region. While the original NS algorithm compared all elements of set *S* to the detector of set *R*, here the same comparison occurs over the period *T*, as the full set of *S* is not known at any given time. The original NS algorithm, as described in Section 4.3.1, would delete any matching detector at this stage from the detector set *R*. However, other options are available and investigated in the next section. In the case that an immature detector is not matched during period *T*, it evolves to the mature stage.

A mature detector has a limited lifetime and does not have the option to be changed during this period. If it was not time limited, natural changes to the input set *S* would be classed as abnormalities. This behaviour is correct for a certain amount of time but the AIS should be able to adapt to normal changes during long-term monitoring. Therefore, when $Age_{Detector} > Lifetime_{Detector}$ the detector element is removed from detector set *R* to allow a new detector to take its space. As abnormalities are supposed to be unlikely events, even a mature detector will most likely not be activated. When a detector is activated during its matured phase, the way it is trained, does not automatically point to a problem but indicates unknown data points or unseen behaviour patterns that were

not witnessed during $period\ T + Age_{Detector}$. This information is passed further up to an alarm system, which needs to review the situation, as it can be an indicator for a near health problem.

4.3.5.3 Detector Placement

The literature review did not reveal a simple solution for the design of AIS with a good detector generation method to fully cover the search space. The initial approach was therefore to investigate the means to achieve good coverage of the search space with a reduced overlap of detectors while maintaining AIS adaptability.

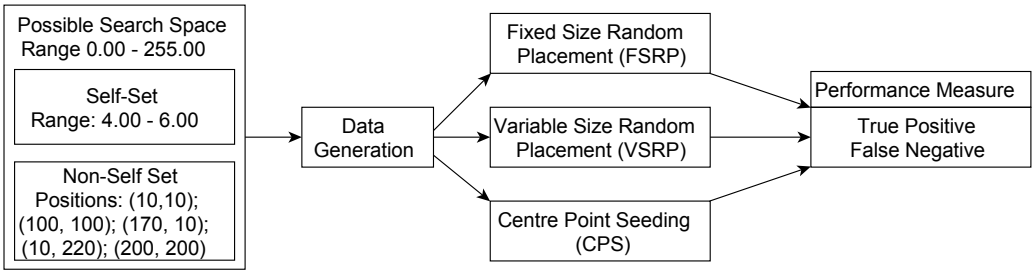


Figure 4–5: Detector evaluation process

Figure 4–5 illustrates the experimental investigation of detector placement that has been carried out and which is described in this section. The investigation is based on two-dimensional data that clearly defines the self-set and non-self set, instead of sensor data collected by the OHDAD, where self and non-self are not well defined. This approach allows the analysis of the different AIS detector placement methods without the influence of possible outlier or unknown sensor data artefact introduced in the data collection. For the two attributes the value range was defined from 0.00 to 255.00 in steps of 0.01 with normal data generated in the range of 4.00 to 6.00 and five abnormal data points at position (10, 10), (100, 100), (170, 10), (10, 220), (200, 200). This combination results in a two-dimensional search space, which has 650250000 possible data combinations. Covering each position with an individual detector would have a sizeable impact on the systems resources and is unnecessary as the affinity area is covering the adjacent area around each detector. Therefore, the amount of detectors is limited to 100 (which is in line with [98]) to limit the system resources needed. The following sections introduce three different detector placement methods.

4.3.5.3.1 Fixed-Size Random Placement

The Fixed-Size Random Placement (FSRP) method places detectors with a fixed affinity area at random positions in the search space. For a possible equal distribution of the

detectors the uniform distribution is used. The Probability Density Function (PDF) of the uniform distribution is presented in equation 4-2 and states that elements between the lower and upper limit have an equal probability to be selected as an attribute value for the DCP, while values outside this range have a probability of 0.

$$f(x|lower, upper) = \begin{cases} (\frac{1}{upper - lower}), & lower \leq x \leq upper \\ 0, & otherwise \end{cases} \quad (4\text{-}2)$$

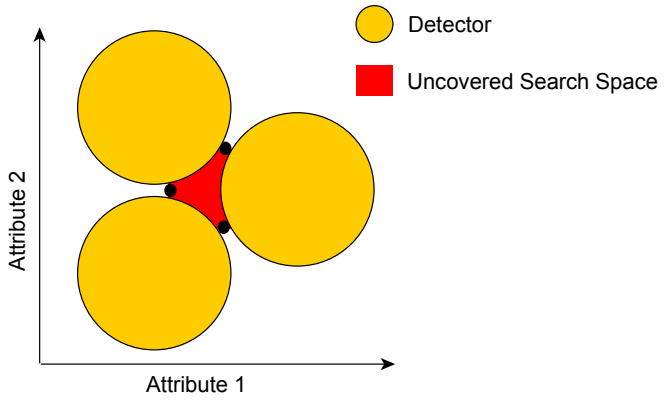


Figure 4–6: Possible uncovered search space area using Fixed-Size Random Placement (FSRP)

A detector is triggered when input data is in range of its affinity area, as described in detail in Section 4.3.5.1. Figure 4–6 illustrates that there exists a trade-off between a large affinity area and the coverable search space. A large affinity radius allows covering wide areas of the search space with fewer detectors, while it can prevent covering areas that lie between at least three data points. A deciding factor is thereby the importance of the uncovered search space and whether additional data points belonging to the self-set or non-self set can be expected within this area. For the experiment described in this section, the generated non-self data is defined to be outside such an area.

Figure 4–7 illustrates the distribution of the DCPs in the hypothetical search space, defined by the generated data; with the affinity area represented as yellow circles. The graphical representation of the distribution allows for a visual analysis of uncovered areas (holes) in the search space by the detectors. In addition, a significant overlap of detector affinity areas is noticeable in some areas, which are the result of detectors being placed too close to each other.

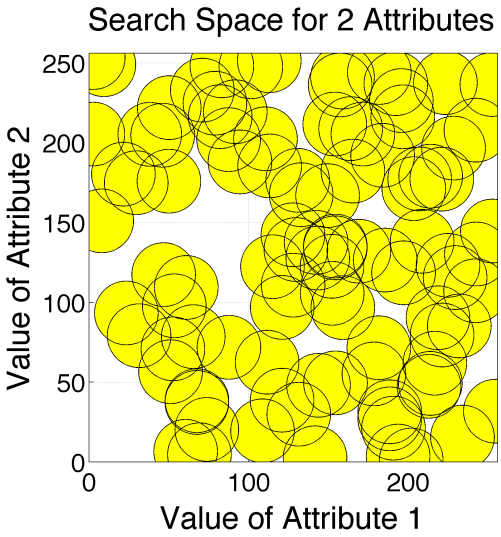


Figure 4–7: Search space coverage using Fixed-Size Random Placement (FSRP)

4.3.5.3.2 Variable-Size Random Placement

The Variable-Size Random Placement (VSRP) method investigated as part of this chapter is similar to the first method. It is also based on randomly placed detectors. The difference between the VSRP and FSRP methods is that the former adopts the following changes in the algorithm to determine the size of the affinity area:

- Start with increased affinity area.
- Let affinity area shrink down to a threshold level.

An increased affinity area increases the probability of a fully covered search space, while the option to shrink the affinity area provides better coverage around normal data points (area of interest), which in this hypothetical scenario lays in the square defined by the four corner points (4, 4), (4, 6), (6, 6), and (6, 4) in the search space.

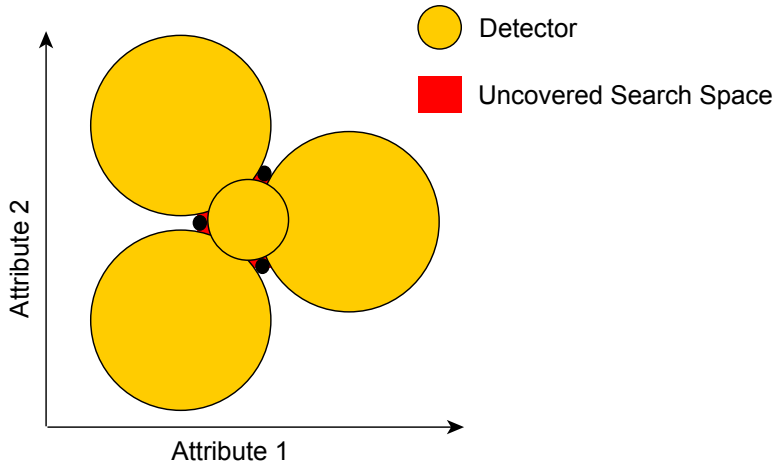


Figure 4–8: Possible covered search space area using Variable-Size Random Placement (VSRP)

The variable affinity area, allows detectors not to be instantaneously removed from the detector set $R_{Immature}$ but can decrease the monitored area around them first before a threshold (MinRadius) defines that the detector was placed too close to normal data. Figure 4–8 illustrates that this approach can also improve the coverage of areas defined by at least three data points, which was not coverable by the earlier described FSRP method.

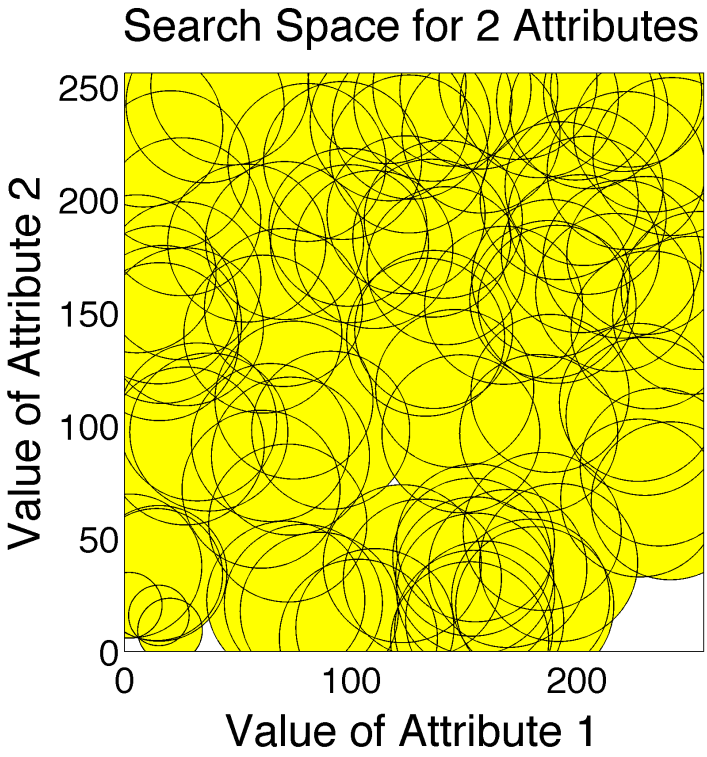


Figure 4–9: Search space coverage using Variable-Size Random Placement (VSRP)

Figure 4–9 illustrates the affinity areas (yellow) around each DCP. The variable affinity area can be seen in the lower left hand corner, where the area of interest lies. Furthermore, the increased affinity area improves the coverage of the search space even though it is still not fully covered. At the same time, the overlap of the affinity areas is increased as well. Therefore, if detectors are placed in a more intelligent way, it will not only reduce overlap but would furthermore help to reduce the amount of detectors needed to cover the complete search space.

4.3.5.3.3 Centre Point Seeding

The apparent limitation of the above methods in terms of coverage led to a third solution to be investigated as part of this work. The Centre Point Seeding (CPS) method places most detectors in the area around the normal data and only a few detectors in the rest of

the search space. Furthermore, the affinity area around a detector should depend on its distance from the area of interest. Consequently, detectors closer to this area should have a small affinity area, while detectors further away should cover a much wider area based on the assumption that data outside the normal area is an infrequent event.

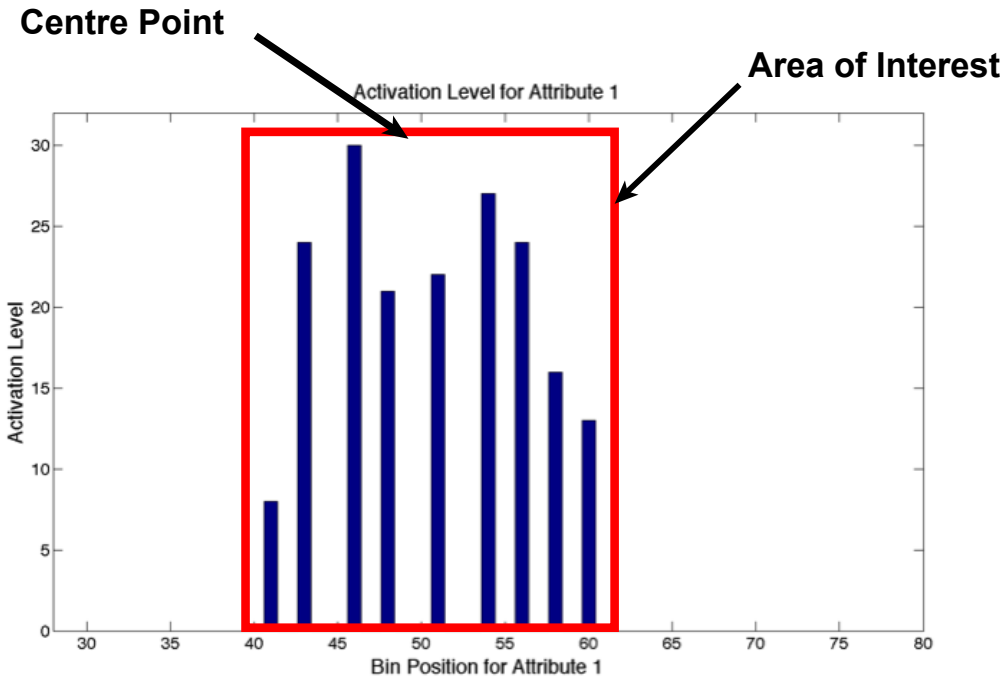


Figure 4–10: Activation level of bins

The author has developed a method to detect the area of interest and start placing detectors around this centre point. The two-step process starts with identifying the area of interest and subsequently placing detectors around this area. Therefore, a given lead-time *x* is required before the identification of the area of interest can begin. In the first stage the continuous input data is discretised. The possible bin value ranges from 0.00 to 255.00 for Attribute 1 and 2. A continuous background process keeps count of each individual input bin (see activation level in Figure 4–10). This input count is later used to detect the possible area of interest, when the activation level of the individual bin is compared to a zero activation level. In Figure 4–10 only one area of interest is visible but additional areas could be at other positions. Therefore the first step is to get all possible areas of interest for each attribute, in a second step these areas can be analysed in more detail to determine their maximum activation level, which translates to the centre point of the activation area. Once these areas are specified, detectors are placed around the centre points. The position of detectors is based on equation 4-3 and equation 4-4.

$$R_x = X_{Detector_{CentrePoint}} + radius * \cos(\vartheta) \qquad (4\text{-}3)$$

$$R_y = Y_{Detector_{CentrePoint}} + radius * \sin(\vartheta) \quad (4\text{-}4)$$

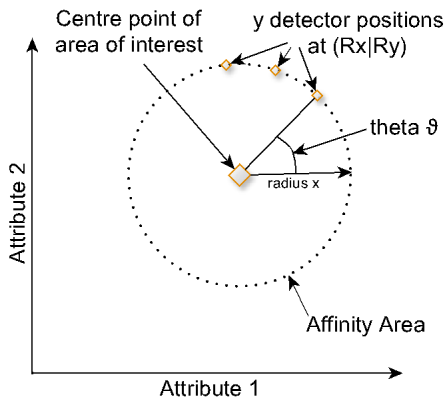


Figure 4–11: Calculation of detector positions

A circle with radius x is divided into y points on the circumference. At every iteration, the radius x is doubled whilst y (the number of points on the perimeter, here 20) stays constant. Therefore the distance between each of the y points is increased. The next step is to use the position $(R_x|R_y)$ of the y points to place detectors (see Figure 4–11). If the detectors were generated with a constant affinity area, this would lead to gaps in the search space. Therefore, the radius of the affinity area is dependent to the distance of the area of interest and calculated using equation 4-5.

$$Radius_{Affinity} = \frac{Radius_{Detector}}{2} + 10 \quad (4\text{-}5)$$

This results in a denser coverage around the area of interest while detectors further out can monitor a wider area. Additionally, this layout prevents the problem of a near 100% overlap between individual detectors that is present in the FSRP and VSRP positioning methods. The CPS method distributes the overlap more equally over the whole search space. As a result less, detectors are needed for the same coverage (see Figure 4–12 showing only the detector positions and Figure 4–13 showing the detector affinity areas around the area of interest).

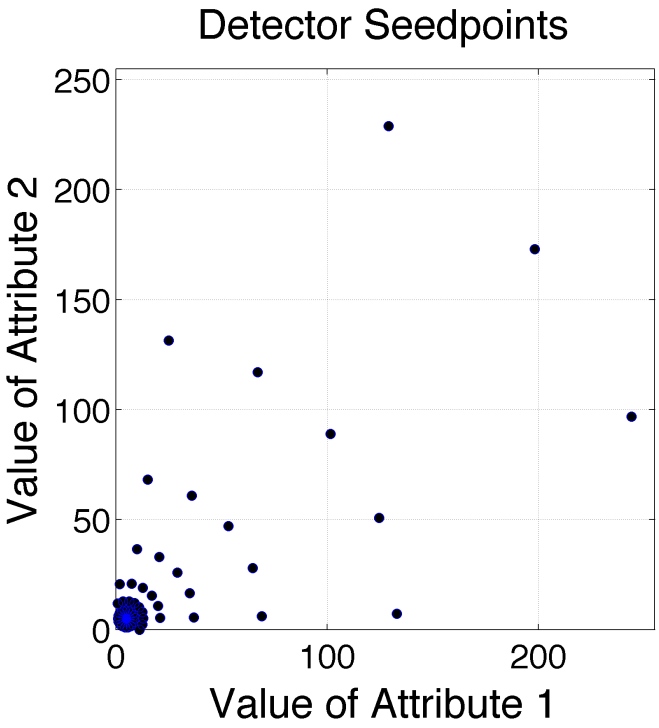


Figure 4–12: Detector centre point placement in the search space

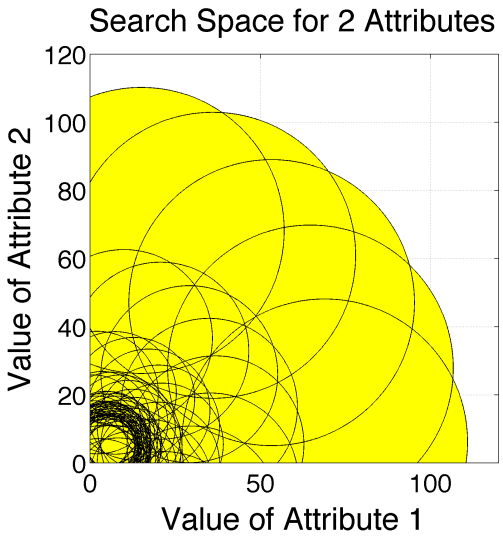


Figure 4–13: Detector affinity area in the search space

4.3.5.3.4 Testing and Analysis of Seeding Methods

The evaluation of the three detector seeding methods introduced in Section 4.3.5.3.1 - 4.3.5.3.3 for their detection of abnormality has been carried out using the generated data as defined in Section 4.3.5.3. The generated two-dimensional search space had 650 250 000 possible data combinations, as the value range for each attribute was defined from 0.00 to 255.00 in steps of 0.01. Normal data was generated in the range of 4.00 to 6.00

and five abnormal data points defined at position (10, 10), (100, 100), (170, 10), (10, 220), (200, 200). A sampling frequency of 0.2 Hz was emulated, with an abnormal data point every 4 days. The author analysed 20 different simulations each lasting 20 simulated days. The performance measures of interest were:

- Correct classification of abnormal data point
- Missing abnormal data point
- Incorrect classification of normal data point

Table 4–2: Simulation results of FSRP, VSRP, and CPS

	FSRP Fixed-Size Random Placement	VSRP Variable-Size Random Placement	CPS Centre Point Seeding
Total	100	100	100
Correct	30	34	100
Missing	70	66	0
Incorrect	0	0	0

Table 4–2 presents the three different performance results from the simulation runs. The table shows that the FSRP and VSRP methods achieve nearly the same results, with the VSRP method showing a slight improvement. The cause of missed abnormal data points is uncovered search space. The number of missed abnormal data points is not constant, as the coverage of detectors changes based on different positions generated by the uniform distribution at each simulation runtime.

On the other side, the placement of detectors using the CPS method achieved a 100% detection rate. The results also highlight that none of the methods alarm incorrectly, hence preventing false alarms, a desirable feature in AAL.

The foundation for good anomaly detection is a fully covered search space and becomes one of the key aspects for AIS in the area of AAL. Therefore, placing detectors using a more intelligent approach around the area of interest, the system achieves an increased confidence of detecting abnormal data points. Also the self-learning feature of AIS proves to be promising and could lead to an elderly monitoring system that automatically adapts to its owner's normal behaviour without costly data collection and annotation beforehand.

4.4 Design of a Monitoring System based on the Artificial Immune System in Combination with the Negative Selection Approach

4.4.1 Motivation

The section above introduced the concept of the AIS in combination with the NS approach. However, as illustrated in Figure 4–14, further steps are necessary for the detection of abnormal activity in AAL. The five steps (Application Domain, Immune Entities, Representation, Affinity Measure, and Immune Algorithm) were presented in [161] as a guideline to reach solutions for generic application problems using Immune Intelligence.

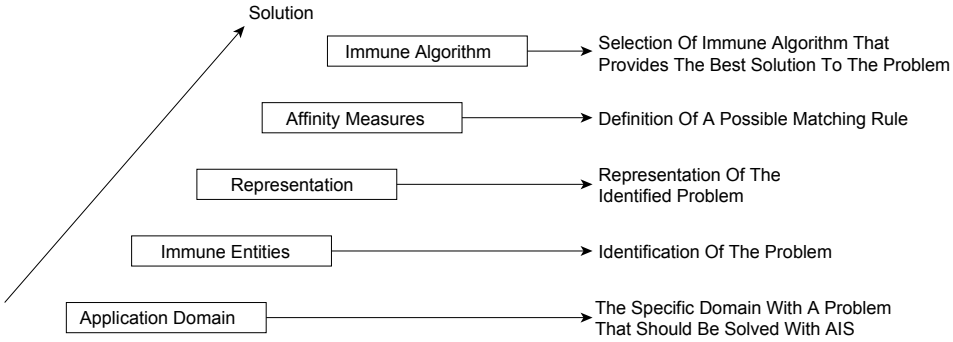


Figure 4–14: AIS solution approach (Adapted from: [161])

The five steps in relation to AAL are:

- Application Domain: For AAL the problem that should be solved using AIS is the monitoring of elderly in a smart environment.
- Immune Entities: The specific problem in the elderly monitoring is the detection of abnormal activity. Different sensors produce various data outputs (see Section 4.4.2) that are required for different aspects of the monitoring of elderly. Section 4.4.2 highlights the use of room occupancy using a PIR sensor and the human activity using an accelerometer sensor.
- Representation: Depending on the sensor data used, different abnormal activities can be monitored. The data representation is determined by the sensor used, in order to allow the differentiation between an AIS self and non-self set (see Section 4.4.2).
- Affinity Measures: Defines the matching rule used by the AIS algorithm to determine the distance of detector and sensor data point, e.g. r-contiguous (see Section 4.3.1) or real valued matching (see Section 4.3.5.1).

- Immune Algorithm: Defines the AIS algorithm that provides the best detection of the abnormal data in the problem representation in the sensor data (see Section 4.4.3.2). Possible AIS algorithms were covered in Table 4–1.

As this is the first time, that AIS in combination with NS is used for the detection of abnormal activity of a person, no prior literature exists. The only literature the author found with regards towards fall classification is from [96] and uses the clonal selection algorithm (CLONALG) for the classification of results of the Morse scale [162], [163] for the fall risk evaluation. Therefore the next section will present possible immune entities, their representation and the monitoring system based on AIS.

4.4.2 Immune Entities and their Representation

Three sections from the above guideline used to find a solution using Immune Intelligence, where not yet covered. Section 4.4.3.2 covers the developed Immune Algorithm, while this section will describe three Immune Entities and their possible representation. The entities were inferred from the literature in Chapter 2 and are limited to room occupancy, raw sensor data, and Fast-Fourier Transform (FFT).

4.4.2.1 Room Occupancy

The common method to passively detect abnormal human behaviour in a domestic environment is the use of Passive InfraRed (PIR) sensors. The room occupancy can be used as a metric to determine over or under room usage, such as found in [68] and [86]. The representation of such problem is illustrated in Figure 4–15, where each hour is represented as an element of data sets S and S^*, which would be compared to detector sets $R_{Immature}$ and R. An AIS system per room would monitor the PIR sensor activation and alarm by over or under usage. If only severe changes in the room occupancy should result in an alarm, the introduction of a self-radius (see Section 4.3.4) should introduce some variation that can be measured in minutes. The disadvantage of this entity and representation is a slow response time to a deviation of normality. One reason is in the nature of the monitoring (passive) and the second is the time that needs to pass before the necessary room occupancy data can be generated.

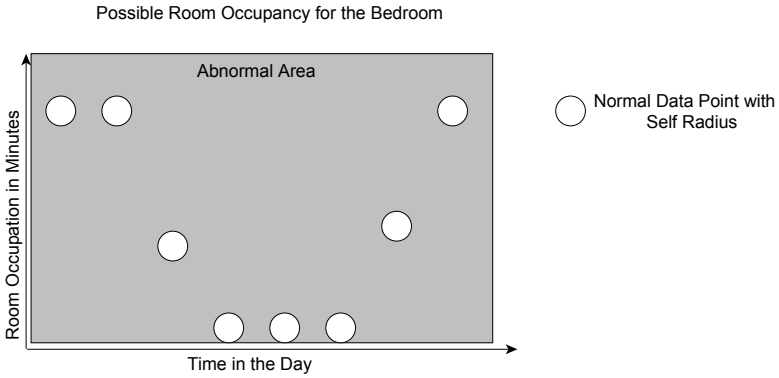


Figure 4–15: Room occupancy presentation

4.4.2.2 Raw Sensor Data

An alternative to passive monitoring is the use of on-body sensors that provide "active" information and allow for a faster response time to problems with the disadvantage of producing significantly more data. The idea of using raw accelerometer information data for the detection of abnormal activity was proposed by the author in [97], with the interest of finding unusually high acceleration peaks. The use of raw sensor data is also motivated by the desire to eliminate the data pre-processing, which increases Computational Load (CL) and introduces additional parameters that increase the complexity in achieving satisfying results.

A Fixed-size Non-overlapping Sliding Window (FNSW) can segment the sensor data in manageable data processing chunks. Longer window sizes will reduce CL, as the AIS algorithm is required to run less often, while shorter window sizes will increase CL but reduce response time. Response times in seconds are desirable, however a trade-off between window size and response time might be necessary to implement a usable solution.

For each window, the contained data elements would belong to the self-set S for detectors of the set $R_{Immature}$ and to the data set S^*, which needs to be classified as self or non-self (see Section 4.3.1), for detectors of the set R.

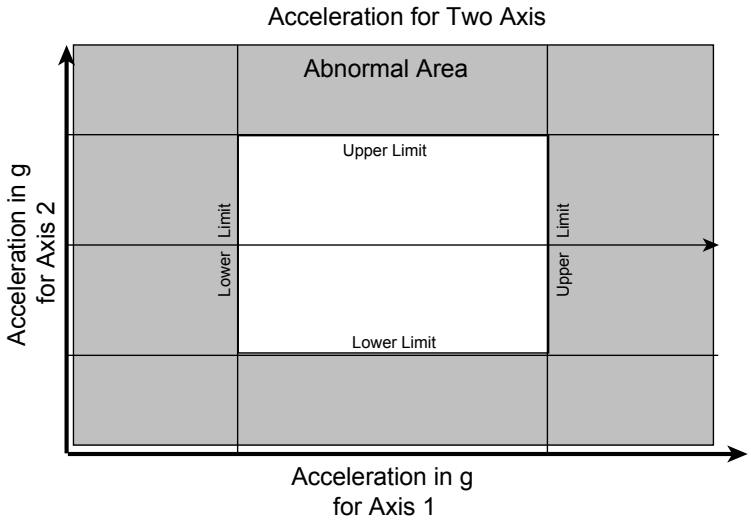


Figure 4–16: Two-dimensional accelerometer presentation

To be able to do this, the acquired on-body accelerometer data needs to be presented to the AIS in a suitable manner. The available OHDAD sensor data includes the X, Y, and Z-axis of the accelerometer (with further data available but not of interest for the work described in this thesis), which can be presented in three different approaches. The first one is on an individual basis, analysing how the acceleration data is distributed in the segmentation window (one-dimensional, 1D, for a segmentation window of one sample). The second approach is illustrated in Figure 4–16 and combines the accelerometer values of two out of the three axes (two-dimensional, 2D, approach). For each accelerometer axis there exists a lower and upper value that indicates normal activity, which will be defined by the AIS. The third approach builds upon the second one and uses all three available axes (three-dimensional, 3D, approach). The first approach is the only one, where the search space is variable and depends on the length of the used window size. Moreover, it should not be assumed that further data points in the immediate proximity belong to the same data point, as it is important not to miss abnormalities. Hence an aggressive self-set should be pursued, with a self-radius of 0 (see Section 4.3.4).

4.4.2.3 Fast Fourier Transformation

The last described approach is a modification of the raw sensor approach proposed above. In [114] the author hypothesised that changes in the activity execution, or sudden movements such as a fall, could possibly be highlighted by changes in the frequency domain characteristics of the data signal. For example, Figure 4–17 illustrates the

transition from standing to falling for 10 Hz sensor data. The FFT was calculated from a 5 s window with a Fixed-sized Overlapping Sliding Window (FOSW) with 90% overlap.

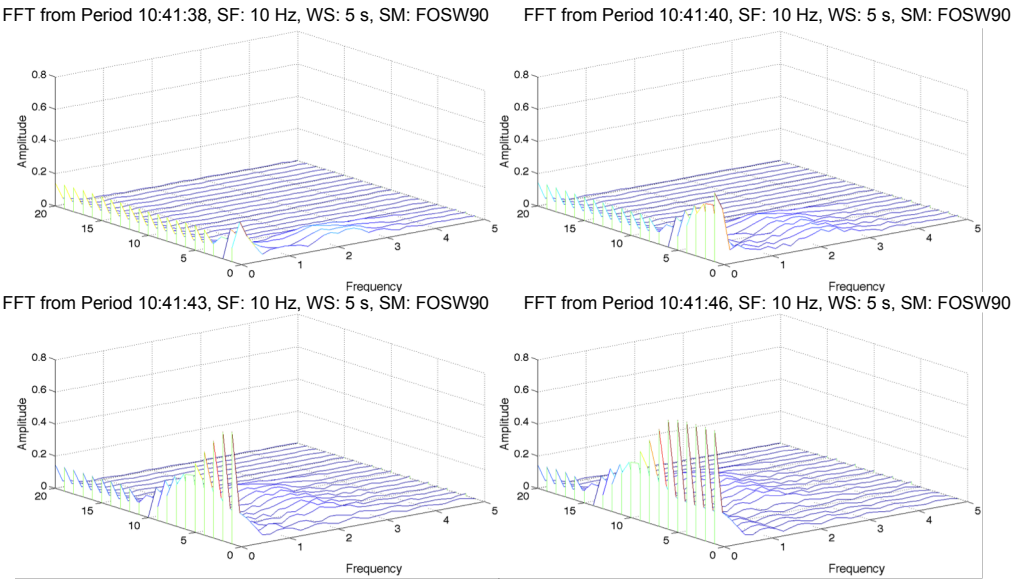


Figure 4–17: Frequency representation of a fall

The transition from time to frequency domain is a two-step process, which begins with the segmentation of the sensor data into manageable processing chunks. Segmentation methods such as those discussed in the literature review in Chapter 2 can be used, e.g. Fixed-size Non-overlapping Sliding Window (FNSW) or Fixed-size Overlapping Sliding Window (FOSW). For both methods the optimal window size would need to be determined, with the latter having the overlap percentage as another adjustable parameter to be determined. Part two, would be the Fast-Fourier Transformation (FFT) of the data window. The resulting outcome could be monitored using AIS. Identical to the raw sensor presentation above, data points in close proximity to each other in the frequency domain should be treated as individual points. Using the aggressive presentation without a self-radius, the AIS can identify abnormal data points, which only slightly deviate from the norm.

4.4.3 Monitoring Abnormal Activity Based on AIS

From Dasgupta's et al. guideline (see Section 4.4.1) that highlighted the five required steps to identify a solution using Immune Intelligence as Application Domain, Immune Entities, Representation, Affinity Measure, and Immune Algorithm, only the latter has not yet been covered apart from a general introduction in Section 4.3.2 for the AIS and NS method.

From the three immune entities and their presentations introduced in Section 4.4.2, only the raw sensor data representation is explored. This limitation is for three reasons. Firstly, it is motivated by the desire to reduce the computational load through the elimination of data pre-processing, such as required by the FFT. The second reason is the expectation to give the best direct feedback in case of an abnormality such as high acceleration peaks or different walking pattern. Thirdly, the results in the author's publication [114] did not highlight significant frequency peaks in the frequency domain. Furthermore, as AIS is a novel approach for the detection of abnormal human activity, using the raw accelerometer data reduces the complexity of the whole system in case of problems. The author further limits the raw sensor data representation to the two-dimensional and three-dimensional approaches that were introduced in Section 4.4.2.2 and use the accelerometer sensor values as axes attributes. This limitation allows for a consistent size of the search space that is independent of the used segmentation window size.

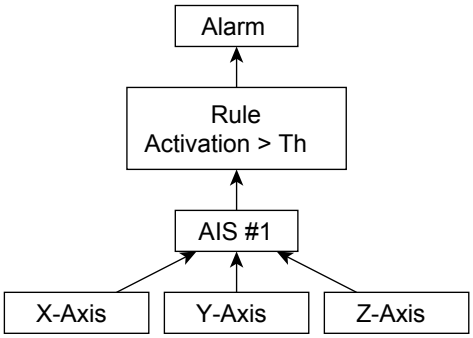


Figure 4–18: Rule-based monitoring system for 3D representation of accelerometer data

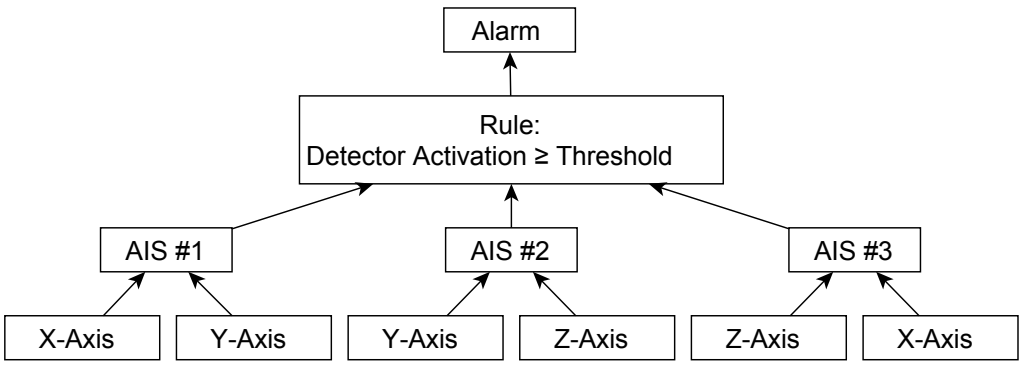


Figure 4–19: Rule-based monitoring system for 2D representation of accelerometer data

The two monitoring system, illustrated in Figure 4–18 and Figure 4–19 for the 3D and 2D data representations were designed and developed, as part of the work presented in this thesis, as it is unknown which combination of accelerometer axes is best for the

monitoring process. Part of the design approach is based on the Hierarchical Temporal Memory (HTM) introduced by Jeff Hawkins [164]. HTM and AIS are both used to generalise the underlying data and only relay abnormal data to higher levels of the monitoring system. In the 3D representation, the X-, Y-, and Z-axes are simultaneously monitored and in the 2D representation the combinations of the three-axes, XY, YZ, and ZX, are monitored in parallel. Both abnormal activity monitoring systems are based on a combination of AIS and rule-based system. The AIS algorithm as the Immune intelligence identifies abnormal data patterns through the activation of detectors and the rule-based system defines an alarm condition based on the AIS detector activation level. Both application blocks are described in detail in a top-down approach in the following sections.

4.4.3.1 Implementation of the Threshold Block

The first application block to be described is the threshold algorithm that determines when the underlying AISs discover an outlier in the data stream or a genuine abnormal activity.

For the threshold algorithm (top level), the function of the AIS block can be seen as a generalisation of the underlying sensor data (bottom level). A change of the bottom level that results in an activation of the AIS does not necessarily represent an event that requires further attention. Therefore a two-step process is defined for the final alarm generation. First, it is hypothesised that an abnormal activity such as a fall will activate more than one AIS detector ($Alarm_{Intensity}$). If the alarm signal of a given AIS block should be treated as an actual alarm ($ActivationLevel = 1$) or a data outlier ($ActivationLevel = 0$) is determined using the $Alarm_{Intensity}$ and equation 4-6.

$$ActivationLevel_{AIS_i}(x) = \begin{cases} 1, & Alarm_{Intensity} \geq Threshold \\ 0, & Alarm_{Intensity} < Threshold \end{cases} \quad (4\text{-}6)$$

In the second stage, the resulting $ActivationLevels$ are used to classify the analysed FNSW data segment as belonging to an alarm condition and escalate the abnormality to the care personnel on call or a more specialised software agent to determine if it is necessary to investigate the cause of the alarm condition. The alarm condition is identified using equation 4-7, where each $ActivationLevel$ of the used AIS blocks is logical AND connected (multiplied). For the 2D representation, this step means that an

alarm can only be triggered when abnormal data is detected in all three accelerometer axes at the same time.

$$Alarm = \prod_{i=1}^{n} ActivationLevel_{AIS_i} \quad (4\text{-}7)$$

with n being the number of AIS system blocks

4.4.3.2 Implementation of the AIS Block

The section above described how the threshold block is used to classify possible alarm conditions from the AIS into outlier and genuine alarm. This section will describe the implemented AIS algorithm and two processes carried out to optimise the detector placement during the AIS runtime.

The framework proposed in [98], which was presented in Section 4.3.2 and introduced the use of a lifespan for AIS detectors to allow the presentation of the self-set *S* over a period of time, is used in a slightly modified version as the base for the AIS algorithm. The implemented life stages for an AIS detector are illustrated in Figure 4–20. Every new detector is randomly generated using the uniform distribution with the PDF presented in equation 4-2. The characteristic of the PDF from the chosen distribution defines that each value between the defined lower and upper limit of the possible value range has an equal probability of being selected as an attribute defining a part of the detector position in the search space. After the generation of a detector, it is placed into the detector set $R_{Immature}$, from where it can mature and move to the detector set *R*, if it is not activated over a defined period *T.* Otherwise it is removed and replaced by a new immature detector. Only as part of the detector set *R* is it possible to trigger an alarm condition, which is in line with the original sense of the Negative Selection (NS) method. In addition, to prevent a persistent coverage of sensor data that can dynamically change, detectors in set *R* are removed after their age exceeds a pre-defined lifetime.

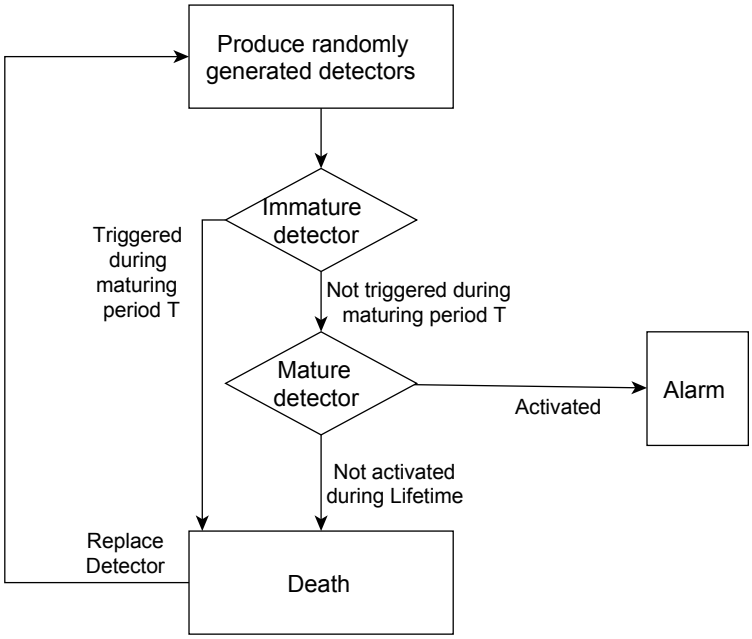


Figure 4–20: Detector life stages based on [98]

Furthermore, Section 4.3.1 pointed out that the important data sets for the AIS algorithms are:

- Self-set S
  - Ideally containing only normal data points.
  - Used for the training of immature detectors belonging to the detector set $R_{Immature}$.
- Data-set S^*
  - Can contain normal as well as abnormal data points
  - Used for the test of mature detectors belonging to the detector set R.

However, using the approach presented by Hofmeyr et al. self-set S and data set S^* are represented by the same data set, where its function is determined by the detector set ($R_{Immature}$ or R) used to calculate the affinity distance. After the possible life stages of a detector and the required terminology, with regards to the different data sets are defined, the general AIS flow diagram can be presented. Figure 4–21 illustrates the AIS function block that is designed to carry out the generalisation of sensor data for as long as data is available.

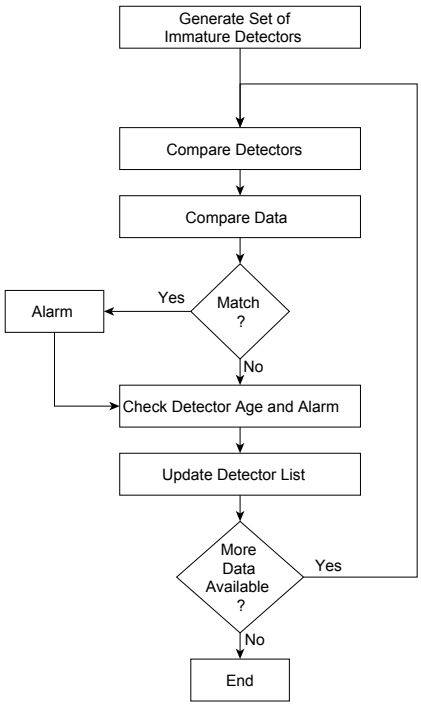


Figure 4–21: Flow diagram of the implemented AIS algorithm

The implemented AIS algorithm is based upon the following five tasks:

- Generate an initial detector set $R_{Immature}$.
- Compare affinity areas between detectors in set $R_{Immature}$.
- Compare distance between data set S and $R_{Immature}$, and compare distance between data set S^* and R.
- Check the age of detectors belonging to set $R_{Immature}$ and R.
- Repopulate detector set $R_{Immature}$ with new immature detectors.

The first task of the AIS is the detector generation for the detector set $R_{Immature}$. During the initial research of the detector placement methods in Section 4.3.5.3, the three different approaches that were investigated were, Fixed-Size Random Placement (FSRP), Variable-Size Random Placement (VSRP), and Centre Point Seeding (CPS). During the analysis it was shown that the FSRP and VSRP approaches resulted in significant detector overlap because of the generation of several immature detectors in close proximity to each other. This led to uncovered search space with the consequence of missing abnormal data points during the simulation with synthetic data. The developed CPS method prevented the generation of coverage holes, as it placed detectors around identified areas of interest (areas with frequent data points) at different distances with increased detector affinity areas. While this approach showed better results to the FSRP and VSRP methods, CPS would not prevent detector overlap when the search space

would include two or more areas of interest. Therefore, an approach that is related to the VSRP method in combination with an optimisation regarding the detector overlap is preferred as it allows for a more general use of the AIS block, for example when the room occupancy and FFT data representations that were introduced in Section 4.4.2 are of interest. Consequently, the uniform distribution in combination with a variable affinity area is used for the detector generation.

The second task of the AIS is the distance calculation between detectors of set $R_{Immature}$. This task is an addition to the original presented VSRP method in Section 4.3.5.3.2 and has the purpose of optimising the detector placement process through the identification of detectors that are in close proximity to one another or share too much affinity area. The flow diagram for this optimisation approach can be found in Appendix D. For all immature detectors i, the $space_{ij}$, j ={1, 2, …, n-1}, with n being the number of immature detectors, is compared to their respective affinity areas, which can have three possible outcomes:

- Detectors do not fall into each other's affinity area.
- Detectors fall into the minimal affinity area defined by the fixed parameter $MinRadius$ (introduced in Section 4.3.5.3.2, where it defines the lower limit for the variable size of the detector).
- Detectors are in the affinity area but outside the $MinRadius$.

The first outcome requires no further action, while the second outcome results in the removal of the youngest matching detector from set $R_{Immature}$. For the third outcome the algorithm reduces the affinity area of the youngest detector to the calculated $distance_{ij}$ minus a fixed pre-defined margin. This is followed by the calculation of the intersection between the two overlapping areas to determine the percentage of affinity area shared between the two detectors in an attempt to improve the coverage of the search space of an individual detector. The calculation of the circle-circle intersection (2D representation) and sphere-sphere intersection (3D representation) as presented in [165] and [166], respectively is presented in Appendix F. Maximising the impact of the individual detector on the search space coverage (max ($impact_{Detector_i}$)) requires the minimisation of the overlap between individual detectors (min{$overlap_{Detector_{ij}}$}), i being the current immature detector and j = {1, 2, …, n-1}, being the remaining immature detectors. Therefore, if the overlap percentage between two detectors is above a fixed

threshold, the youngest detector is removed from the set $R_{Immature}$, allowing a new immature detector to be generated at a different location with a possible better (max ($impact_{Detector_i}$)).

In the third task of the AIS, the detector elements of the detector set $R_{Immature}$ are compared with the self-set S. If an element S falls into the affinity area of an immature detector, the affinity radius is reduced to the point that no further activation can occur by a self-element at the same position. If this requires a radius below the defined $MinRadius$, the detector is removed from the set $R_{Immature}$. For elements of the mature detector set R, the same matching process is carried out. If part of the sensor data set S^* falls into the affinity area of a mature detector, the possible alarm situation is noted and passed to the next higher hierarchical level (here the threshold block). This process was also described in Section 4.3.5.3.2 for the VSRP approach.

The fourth task is designed to reset possible alarm conditions of detectors and monitors when detectors achieve their next life stage. The former process requires only the reset of an attribute flag of a detector, while the latter requires additional steps. The flow diagram of this function is illustrated in Appendix E.

When an immature detector achieves maturity and is supposed to move from the detector set $R_{Immature}$ to set R, the condition that needs to be fulfilled is that of space availability for an additional element in R. Furthermore, as the AIS is designed for a resource limited environment, new mature detectors should not just be placed into the detector set R, even if space is available. In an optimisation approach, the new mature detector i is compared to the j detectors in set R to determine if one or more detectors might be replaceable. Figure 4–22.a and Figure 4–22.b illustrate the fully and partially overlapping relation between detector i and j, under the condition that the $Distance < Radius_i + Radius_j$. Every detector j that is fully overlapped by detector i is removed from the resource-limited detector set R. The removal of older mature detectors not only frees resource, while keeping the same search space covered but also improves the computational load, as less mature detectors are matched with elements of the continuous data set S^* during the third AIS task. Furthermore, partially overlapping detectors are also integrated in the optimisation process of detector set R. For this the maximum overlap between detectors i and j needs to be determined using equation 4-8 (excluding detector combinations that already resulted in a complete overlap).

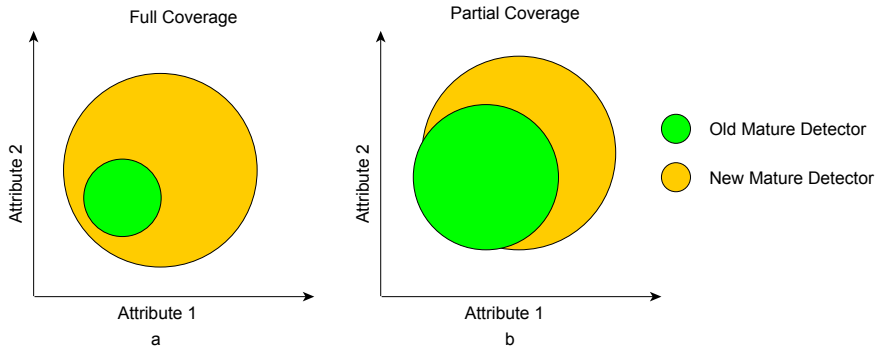


Figure 4–22: Possible relations between old and new mature detector

$$Overlap_{maximum_{ij}} = \max\{Overlap_{i1}, Overlap_{i2}, \dots, Overlap_{ij}\} \quad (4\text{-}8)$$

The detector that has the maximum overlap with detector i is indicated as detector k. This is followed by the calculation of $AffinityArea_k$, to decide if the old mature detector k can be replaced by the new mature detector i. For this equation 4-9 has to be true, whereby the $MatureOverlapPercentage$ is a fixed predefined parameter that determines the percentage of the $AffinityArea_{ik}$ that needs to overlap. Even though some search space coverage might be lost in this process, the majority of the area will be covered for at least another mature detector lifetime.

$$Overlap_{maximum_{ik}} > AffinityArea_k * MatureOverlapPercentage_k \quad (4\text{-}9)$$

The last step of the AIS block before the next data segment is analysed is the repopulation of immature detectors in set $R_{Immature}$. The new immature detectors are generated using the same uniform distribution as described in the initial detector generation process in the first AIS task. The characteristic of the PDF presented in equation 4-2 and defines that each value between the defined lower and upper limit of the possible value range has an equal probability of being selected as an attribute defining a part of the detector position in the search space.

4.4.4 Discussion

The sections above have highlighted the possible usage and convenience the AIS concept provides for the detection of abnormality in AAL environments. In addition, the implementation of an AIS-based monitoring system was presented. Though the question stands, on how does it fit into already existing concepts in the application area? The author choose the complexity of the underlying data model as a way to compare different approaches, as computational time and code complexity cannot be adequately measured for all algorithms. On one side, the AIS approach is more complex than solutions that are

purely based on simple threshold values. These solutions also require a-priori knowledge to determine the threshold values and do not automatically adjust to required changes. On the other hand, the AIS data model is not as complex as traditional supervised classification methods that have a static learned data model and differentiate between different activity classes. The problem with these methods is that extensive data collection is required to gather a-priori knowledge or a given problem needs to be modelled beforehand using expert knowledge. Thereby, it cannot be guaranteed that every possible abnormality is considered with the a-priori and expert knowledge. Therefore, the author believes that AIS should find its place between these two described methods, as illustrated in Figure 4–23.

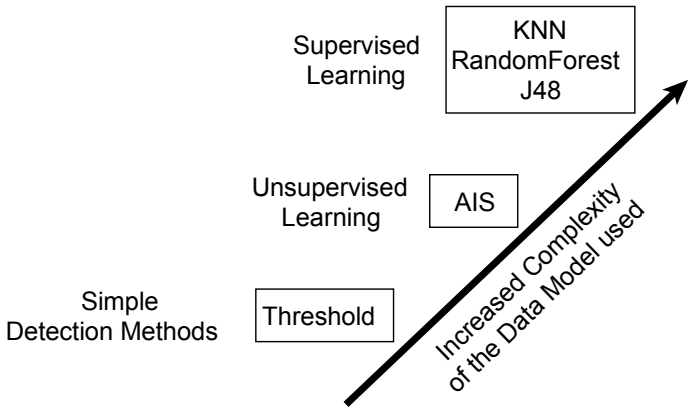


Figure 4–23: Position of AIS in the current work in Ambient Assisted Living

Furthermore, sometimes a deciding factor for the use of a specific classification algorithm can be driven by the need to be able to understand and interpret the results. Therefore, decision trees might be preferred over e.g. Artificial Neural Networks (ANN), if the classification accuracy is similar. Regarding this aspect, the alarm condition of an AIS system represents unusual data activity that was not witnessed for the duration of the immature period T and the age of a mature detector. This information, combined with the last x minutes of vital and activity data of a test subject, might already be enough for the expert knowledge of a care assistant to determine the severity of an event. Similarly, a specialised software agent should not be required to analyse obviously normal data. The monitoring system developed as part of this work, can be used as a triggering system for a more in-depth analysis, resulting in optimised resource usage.

4.5 Summary

The section above introduced the less known concept of Computer Immunology and its resulting AIS abnormality detection algorithm for the area of AAL developed as part of

the work carried out by the author. The different implementation approaches were highlighted, with the detailed explanation of the Negative Selection (NS) algorithm as proposed by [98], which is adaptable over time and therefore better suited than a static approach for the AAL application at hand. The author also highlighted the initial research regarding different detector seeding approaches, which has resulted in a journal publication [167]. Based on this investigation, a novel monitoring system that improved on the detector placement was developed. The core components, including two optimization approaches are described in detail. The application flow diagrams can be found in Appendix D and Appendix E. The validation of the monitoring system will be presented in Chapter 6 alongside other methods investigated in Chapter 5 for comparison purposes.

5 Identification of Good Parameter Combinations for the Evaluation Process of Comparing AIS to Other Supervised Classification Algorithms

5.1 Introduction

This chapter presents an empirical investigation of different data sampling rates, segmentation techniques, and segmentation window sizes and their effect on the accuracy of Activity of Daily Living (ADL) event classification and computational load for two different accelerometer sensor data sets in an attempt to identify possible classification methods and associated parameter combinations for the comparison with the AIS monitoring system implemented in Chapter 4.

Following the motivation, which describes the necessity for this investigation, the experimental procedure that is carried out will be explained in detail. Following this the results of the variation in parameter combinations will be presented with regards to three different analytical aspects. The first two use an ANalysis Of VAriance (ANOVA) to present the effect on classification accuracy and computational load. The third analytical aspect is the combination of classification accuracy and computational load using a Pareto curve based approach. A summary, discussion of the results and conclusion will finish this chapter. The work presented in this chapter appeared as a peer-reviewed publication in Sensor's special issue on Ambient Assisted Living (AAL): Sensors, Architectures and Applications and can be accessed from [168].

5.2 Motivation

This section attempts to identify suitable classification algorithms and their pre-processing parameter combinations, sampling frequency (SF) segmentation method (SM) and window size (WS) in order to compare the developed AIS-based monitoring algorithm from Chapter 4 with other established methods identified in the literature. Since Activity of Daily Living (ADL) events and falls are classified based on features calculated from monitoring sensor data fragments, the selection of good pre-processing parameter have a significant impact on the event classification accuracy (Pietka, 1988). Even though researchers are aware of the importance of sampling frequency (SF), segmentation method (SM), and window size (WS), with respect to feature extraction and classification accuracy, the issue is not addressed in the reviewed studies with no

clear explanation or justification given for the parameter selections. Table 5–1 is a summary of the inconsistent parameter combinations, highlighted by the literature review in Chapter 2. For example, the sampling frequency ranges from 5 Hz as the lowest used value, up to 512 Hz, with a significant amount of work concentrating on values around 50 Hz. Further inconsistency is observable for the used window sizes with values starting as low as 0.25 s up to 74 s. The only parameter that seams to share a common value for most of the reviewed studies is the segmentation method with a Fixed-size Overlapping Sliding Window (FOSW) with 50% overlap, however the reason for this specific parameter value is not explained in the literature.

Table 5–1: Found inconsistency in sampling rate, segmentation method, and segmentation window for Ambient Assisted Living

Authors	Sampling Frequency [Hz]	Segmentation Window [s]	Segmentation Method	Classifier Method
Huynh et al., 2005 [37]	512	0.25, 0.5, 1, 2, 4	FNSW, FOSW 50%, FOSW 75%, FOSW 80.5%, FOSW 93.75%	K-Means
Sekine et al., 2000 [38]	256			
Bao et al., 2004 [25]	76.25	6.7	FOSW 50%	DecisionTable, KNN, DecisionTree, NaïveBayes
Preece et al., 2009 [40]	64	2 and 3	FOSW 50%	KNN
Wang et al., 2007 [41]	50	2.56	FOSW 50%	Multi-layer Perceptron Neural Network
Casale et al., 2011 [31]	52	1	FOSW 50%	RandomForest
Ravi et al., 2005 [42]	50	5.12	FOSW 50%	DecisionTables, DecisionTree, KNN, SVM, Naïve Bayes, Boosting, Bagging, Plurality Voting
Pärkkä et al., 2010 [43]	50	5		Binary DecisionTree
Maurer et al., 2006 [44]	50	4	FOSW 92%	DecisionTree, KNN, Naïve Bayes
Antonsson et al., 1985 [45]	1 - 30			
Bouten et al., 1997 [46]	20			
Gjoreski et al., 2010 [47]	5	1.4		DecisionTree, RandomForest
Nyan et al., 2006 [53]	256	2		Threshold
Kasteren et al., 2008 [54]		60	FNSW	
Patterson et al., 2005 [55]		74		
Pietka et al., 1988 [48]			FNSW, FOSW, SAX, SM	
Keogh et al., 2001 [49]			FNSW, FOSW, Bup, SWAB	
Chu et al., 1995 [50]			RbW	
Kozina et al., 2011 [51]			Dwin	RandomForest
Ortiz Laguna et al., 2011 [52]			VSW	DecisionTree, Dynamic Bayesian Network, Part

The above variation in parameter combinations represents the obvious absence of clear instructions regarding the selection of parameters in the field of Ambient Assisted Living (AAL) for the ADL event classification. Furthermore, only a few studies report the variation of parameter value combinations in an attempt to improve classification accuracy. Another factor that tends to be ignored in the literature is the required computational load, which is of particular interest when data classification takes place on an embedded system for real time ADL recognition.

The various different parameter combinations found in the literature, do not allow the selection and justification of good classification methods and their corresponding parameter combinations. However, this is a requirement, when the classification performance to detect abnormal activity using the AIS-based monitoring systems that were developed in Chapter 4, should be compared to other classification algorithms. Another aspect to consider is the expectable loss in classification accuracy, as well as the general validity of the OHDAD for the data collection, with most studies having sampling rates above the achievable 10 Hz.

It is therefore necessary to empirically investigate the influence of the data sampling rate, segmentation techniques, and segmentation window sizes and their effect on accuracy of ADL event classification and computational load. The analysis of various parameter combinations should provide a more informed approach to parameter selection for the comparison with the AIS-based monitoring systems. The analysis of different sampling rates should also provide a better understanding of the limitations introduced when using the OHDAD.

5.3 Investigation Procedure

Figure 5–1 illustrates the investigation procedure carried out in this chapter. Two different data sets found in the literature, containing accelerometer sensor information of different test subjects, are classified using various parameter combinations of Sampling Frequency (SF), Segmentation Method (SM), Window Size (WS), and Classifier Method (CM). Using Matlab [169], sensor data was resampled, segmented and relevant data features (see Extracted Sensor Features box in Figure 5–1) extracted, while the Weka software package [170] provided the implementation of the classification algorithms. The performance measures of interest are Classification Accuracy (CA) and Computational Load (CL). The resulting information was analysed using an ANalysis Of

VAriance (ANOVA), provided by SPSS from IBM [171], for each individual performance measure and using the Pareto curve for a visual identification of best parameter combinations for the combined performance measures.

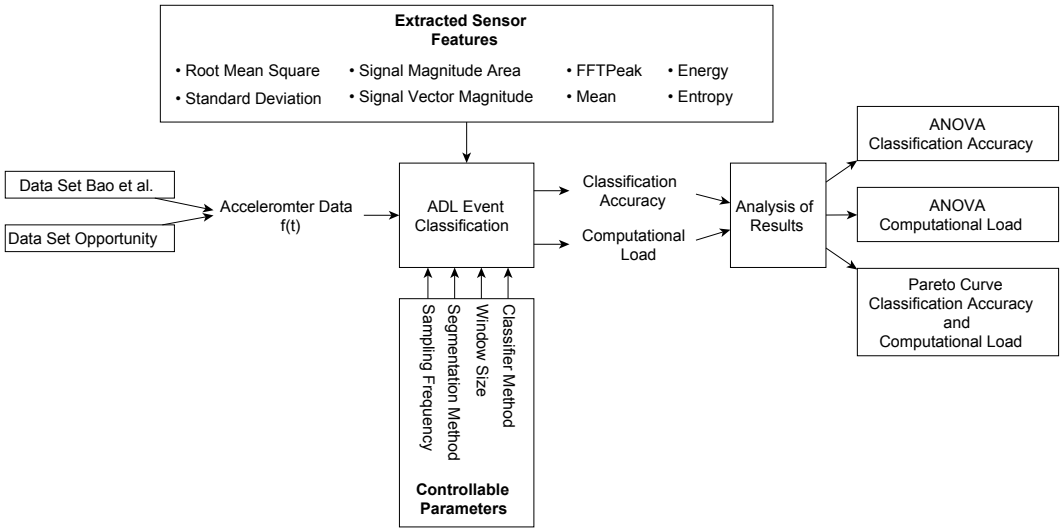


Figure 5–1: Investigation procedure to determine good parameter combinations

5.3.1 Data Sets

The first data set used as part of this part of the work contains bi-axial accelerometer data collected by [25]. The data was generated from 20 different participants (13 males and 7 females with a mean age of 21.8 years (± 6.59 years standard deviation)), which were recruited at MIT with the help of posters. The experiment required the test subjects to execute several different ADLs under laboratory conditions without any supervision or guidance. Sensor data was collected simultaneously at 5 different positions (ankle, thigh, wrist, hip, upper arm), with a sampling frequency of 76.25 Hz. From the five sensor positions, the data of the hip sensor from all twenty different participants is used. The focus is on ADLs such as, walking, sitting, walking and carrying an item, standing still, lying down, and climbing stairs, as they are close to the activities (walking, standing, sitting, and lying) introduced in Chapter 3 and later used in Chapter 6 for the evaluation of the AIS-based monitoring algorithm.

The second data set is the Opportunity data set collected as part of a European Funded project by [23]. The data set is not limited to just body-worn accelerometer data. The complete set includes a total of 10 different sensor types, such as microphone, magnetometer, Ultra Wide Band (UWB) localization, Radio Frequency IDentification (RFID) etc., totalling a collection of 72 sensors. Data was recorded with 12 test subjects, which are not further specified. Of these 12 subjects only 3 subjects are labelled and

available in the University of California Irvine Machine Learning Repository (UCI MLR) [22]. The four locomotion activities that are labelled in this data set are stand, walk, sit, and lie. The Opportunity data set was limited to the accelerometer (sampled at 64 Hz) placed at the subject's hip and only the X- and Y-axes were used, as the investigation for both data sets should be similar to allow for comparable results.

5.3.2 Resampling

Table 5–1 in Section 5.2 above, showed that sampling rates vary greatly throughout the literature; it also indicated a high use of sampling rates around 50 Hz even though Maurer et al. argue that sampling rates above 20 Hz only marginally improve the classification accuracy [44]. Therefore the data sets were resampled using Matlab at six different sampling rates in the range of 10 Hz to 60 Hz in 10 Hz steps. Intermediate steps were ignored for the benefit of faster experiments, as well as the author's belief that the omission of intermediate steps would not cause loss of generality of the results. Additionally, sampling rates above 60 Hz were excluded, as the author agrees with [40], who stated that higher sampling rates are harder to achieve with off-the-shelf components, as seen by the 10 Hz limitation by the OHDAD and TI eZ430-Chronos [113], both introduced in Chapter 3.

5.3.3 Data Segmentation

The work presented here focuses on three online segmentation techniques: Fixed-size Non-overlapping Sliding Window (FNSW), Fixed-size Overlapping Sliding Window (FOSW) (with 4 different overlap percentages), and Sliding Window And Bottom-up (SWAB) that were introduced in the literature review in Chapter 2. The advantages of these algorithms are that they are online capable, therefore can be used while the data collection is in progress and are simple and intuitive so that they are easily understood. Furthermore, FNSW and FOSW are the most common static segmentation algorithms used in literature (see Table 5–1 in Section 5.2). In addition, a dynamic segmentation algorithm (SWAB) allows the investigation of a possible performance improvement over the two included static approaches as reported in [52].

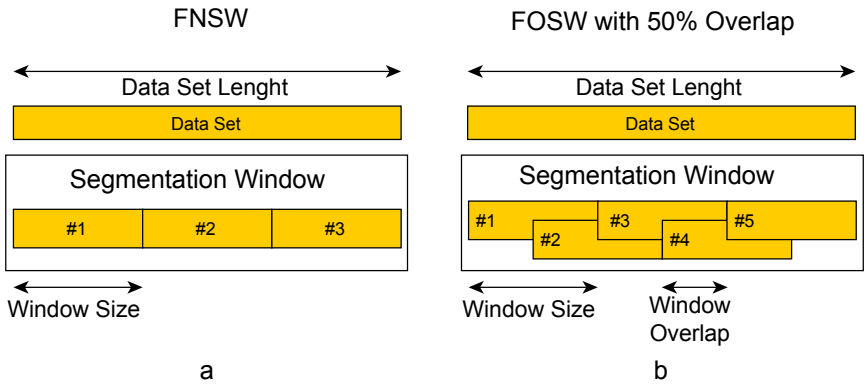


Figure 5–2: Explanation of segmentation methods FNSW and FOSW with 50% overlap

The first segmentation technique is FNSW without any data overlap (see Figure 5–2.a). The end point of segmentation window N is the starting point for window N+1. It is therefore possible to calculate exactly the number of windows generated for a given data set size with equation 5-1:

$$Segmentation_{Windows} = \frac{S}{L_i} \tag{5-1}$$

where S is the total number of signal samples and $L_i = R_{Sampling} * Window_{Size}$, where $R_{Sampling}$ is the data resampling rate used (in the range of 10 Hz to 60 Hz) and $Window_{Size}$ is the selected window size (in the range of 0.5 s to 24 s). One disadvantage of this technique is that data associated with a particular feature (e.g. fall) can be split between windows. A FNSW sliding technique that is not covered in this work is to leave a gap between adjacent windows, as this would result in uncovered sensor data and therefore could miss important information.

The second technique is FOSW and based on FNSW with the addition of data overlap. Figure 5–2.b illustrates FOSW with an overlap of 50%. Depending on the percentage overlap, more or less data overlaps from window N into N+1. This is also referred to as a window shift. A 0% overlap corresponds to the FNSW segmentation method, while an overlap of 100% would yield to a static window as it would not be shifted and the data would always be segmented at the exact same point. Therefore, the requirement for FOSW is to move by at least one data point per segmentation. The number of segmentation windows generated can be calculated using equation 5-2:

$$Segmentation_{Windows} = \frac{S}{L_i - L_i * \frac{P_j}{100}} \tag{5-2}$$

with P_j being one of the following percentage overlap values used for this research: 25, 50, 75, and 90 because they can be found in Table 5–1 in Section 5.2. The two common overlap values of 0% (FNSW) and 50%, resulted in the interest of the author to see if an intermediate step of 25% would show an accuracy increase similar to the FNSW method, FOSW with 50% overlap, or actually in between.

SWAB is the third segmentation technique used as part of the study presented here and was designed by [49]. It is a combination of the Sliding Window And Bottom-up approach. The process is visualized in Figure 5–3. The algorithm has a fixed size data buffer of the length n that is used for the Bottom Up approximation. The first step is to divide the data buffer into $\frac{n}{2}$ data segments. Each of these segments is associated with a given cost for the connection to the next segment. The Bottom-up algorithm merges the lowest cost segments until only a pre-defined number of segments remain. For this experiment, the stopping condition is defined as 6, as it was highlighted in [49] that the buffer size should be set to create 5 to 6 segments. Once the approximation for the window is complete the data buffer shifts by the first segment (in the illustration identified as Segment #1) and the process is repeated for the new buffer window. Each segment is used for feature extraction. As the data shift is dependent on the data set and its approximation, it is not possible to estimate the amount of segmentation windows generated by the algorithm. The implementation is more complex compared to the FNSW and FOSW methods described above and therefore an increased CL is expected while the accuracy gain is uncertain even though literature suggests improved results.

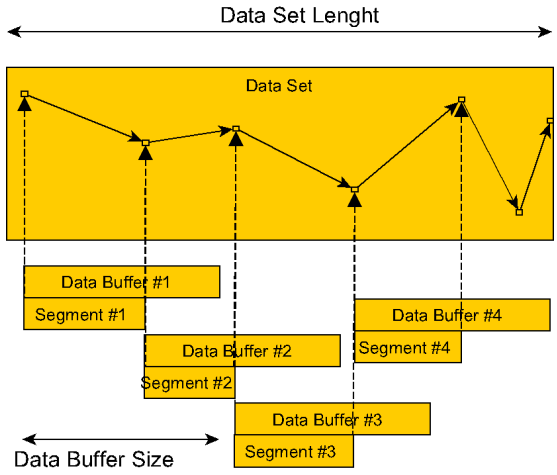


Figure 5–3: Explanation of segmentation method SWAB

As highlighted earlier in Section 5.2, there is no clear recommendation in the published literature on the selection of the window size used for the data segmentation. Therefore,

32 different sizes in the range of 0.5 s to 24 s are tested. In the area of 0.5 s to 8 s, the size is increased in 0.5 s steps, while thereafter the step size is increased to 1 s. The 0.5 s step size was increased after 8 s because the ADLs under investigation have only a short time frame and it allowed the CL to be reduced for the experiment. Even though literature showed the use of longer window sizes the aim is to only include single ADLs in each window to achieve the best classification results. The initial research [172] supports this idea as it indicates a decrease in accuracy for window sizes above 8 s. Furthermore, it was the author's belief that most ADLs will only take a short amount of time and a maximum of 24 s should be sufficient to include at least two ADLs.

5.3.4 Data Feature Selection

The following eight metrics are commonly used in the area of ADL classification and therefore used to retrieve the different features of the accelerometer sensor data in this research: Root Mean Square (RMS), Mean, Signal Magnitude Area (SMA), Signal Vector Magnitude (here SMV), Energy, Entropy, FFTPeak, Standard Deviation (STD). The individual influence of the extracted features is not part of the investigation carried out as part of this thesis. These metrics and their significance are discussed below as each individual metric has its own influence in the research field. For the remainder of the section, n represents the maximum number of elements in a data segment and x_i and y_i indicate the i-th accelerometer value element in the data window for the X- or Y-axis, respectively.

RMS has been used to distinguish walking patterns [39] as well as being an input to classifiers for activity recognition [47], [173]. The RMS value is calculated using equation 5-3:

$$RMS_{x-Axis} = \sqrt{\frac{\sum_{i=1}^{n} {x_i}^2}{n}} \tag{5-3}$$

The Mean metric (see equation 5-4) has been used to: recognize sitting and standing [173], [174]; discriminate between periods of activity and rest [175]; and as an input to classifiers such as Decision Table, KNN, J48, Naïve Bayes, Random Forest, Hidden Markov Model (HMM) [25], [47], [176], [177].

$$Mean_{X-Axis} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{5-4}$$

The next metric, SMA, is used to distinguish between periods of activity and rest in order to identify when the subject is mobilizing and undertaking activities, and when they are immobile [46], [127], [178]. Equation 5-5 implements SMA.

$$SMA = \frac{\sum_{i=1}^{n}(|x_i| + |y_i|)}{n} \tag{5-5}$$

SMV, normally referred to as Signal Vector Magnitude (SVM) but changed to SMV to avoid confusion with the Support Vector Machine (SVM) classifier used, indicates the degree of movement intensity and is an essential metric in fall detection [127], [178]. The SMV value is calculated using equation 5-6.

$$SMV = \frac{\sum_{i=1}^{n}\sqrt{{x_i}^2 + {y_i}^2}}{n} \tag{5-6}$$

Two additional metrics used in this research are Energy and Entropy, which discriminate between types of ADL such as walking, standing still, running, sitting and relaxing [25], [179]. The calculation of the Energy value is based on equation 5-7 and the Entropy is calculated using the Matlab function from [180].

$$Energy_{x-Axis} = \frac{\sum_{i=1}^{n} FFT_{x_i}^2}{n} - Mean_{X-Axis} \tag{5-7}$$

Another feature that was extracted from the accelerometer data stream is the FFTPeak for each axis. The metric has been used for activity recognition [37], [43], [179]. The FFTPeak algorithm was based on the FFT Matlab Example found at [181].

The last metric used is the calculation of the Standard Deviation (STD), which has been extensively used for activity recognition [174]; and as an input to classifiers, such as J48, Random Forest and Artificial Neural Networks (ANN) [47], [182]. Equation 5-8 describes the calculation.

$$STD_{X-Axis} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - Mean_{X-Axis})^2}{n}} \tag{5-8}$$

5.3.5 Classifier Selection

The software tool Weka [170] implements the algorithms of several different classifiers from which nine were selected based on literature to verify the effects of changes in the described parameters above. [43] points out that common algorithms for activity classification are Support Vector Machines (SVM), Decision Trees, and Bayesian

classifiers. The work of [44] included the use of Decision Tress and Naïve Bayes classifier. [31] based their research on Decision Trees, Bagging of 10 Decision Trees, AdaBoost using Decision Trees as base classifiers and a Random Forest of 10 Decision Trees. Additionally, in the work of [47] the authors compare J48 (Decision Trees) and Random Forest. This investigation therefore included the following classifiers: Naïve Bayes, SMO (based on SVM), K-Nearest Neighbour (KNN), KStar, MultiClassClassifier (MCC), Bagging, Decision Table, J48, and Random Forest. All algorithms were tested using Weka's standard configuration and a 10 fold cross validation. Fine-tuning a classifier configuration is a research field in its own right and therefore not discussed here. The use of different classification methods has enabled the author to verify the impact of the SF, SM, and WS on the CA and CL over a wide field of classification algorithms used in AAL. It has also allowed for a wider variety to select possible competing classifier methods.

5.4 Experimental Results

This section will be split in three sub sections to focus on the different aspects of the parameter selection problem introduced by the variation in Classification Methods (CM), Sampling Frequency (SF), Segmentation Methods (SM) and Window Size (WS). The first two analyses use an ANalysis Of VAriance (ANOVA) to investigate the impact of the identified parameters on the two dependent variables: Classification Accuracy (CA, as described in Section 5.4.1) and Computational Load (CL, as described in Section 5.4.2). The analyses have been conducted in SPSS [183]. The software also generated the tables and figures found in these two sub sections. The tables are thereby useful to determine the parameters that have a significant or non-significant role in influencing CA or CL. The figures on the other hand provide a visual guide to help understand how changes in one parameter influence CA or CL for a given second parameter. This allows drawing a conclusion for example if longer or shorter window sizes are better on the CA for the different segmentation methods tested. The third sub section presents a Pareto curve (see Section 5.4.3) that highlights the results of the investigation into the effect various input parameter have on the combination of CA and CL. Moreover, this tool enables the selection of best parameters for the later comparison of abnormality detection. A general summary and discussion of the ANOVA and Pareto curve of the two different data sets is presented in Section 5.5

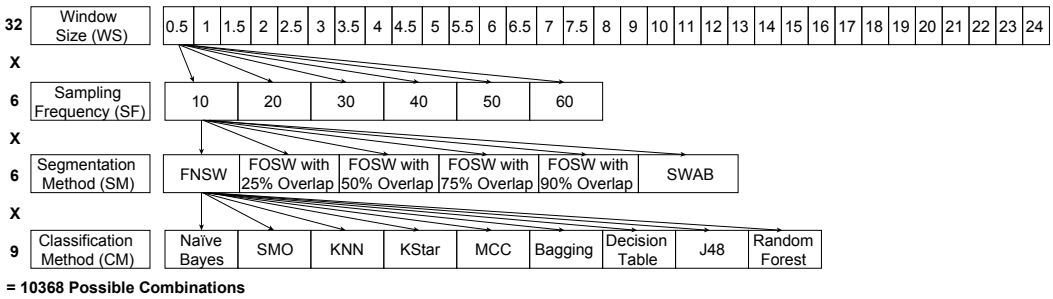


Figure 5–4: Parameter combinations for each test subject

Figure 5–4 illustrates the different levels of the four parameters WS, SF, SM, and CM; this has resulted in 32 different window sizes, 6 sampling frequencies, and three segmentation methods with different parameters (resulting in 6 SM levels) for each of the 9 different classification algorithms. This results in 10368 different parameter levels for each of the 23 tested subjects for the ANalysis Of VAriance (ANOVA) in SPSS. The performance measures used are Classification Accuracy (CA) and Computational Load (CL), as explained in Section 5.4.1 and Section 5.4.2, respectively. As well as classification accuracy, the literature also highlighted the use of precision, recall, and f-measure. In [142] the author argues that precision, recall, and f-measure are especially useful for highly imbalanced data sets. For example, when faced with a two-class problem with a split of 98% (majority) and 2% (minority), just guessing the majority class will achieve an accuracy of 98%. If the detection of the minority class, say, representing rare and infrequent events (e.g. falls), is important, an accuracy of 98% would be misleading in terms of the performance of the classifier. F-measure is thereby used to calculate the harmonic mean between precision and recall. The former determines how often a predicted class label (e.g. fall) for a data segment matched the actual class label, while the latter indicates how many data segments labelled as a specific class were correctly predicted to belong to this class. The two data sets (Bao et al. and Opportunity) used for this research included six and four different activities, respectively. In the data set, the distribution of the different activities was equal and equally important to classify, hence not creating an imbalance, compared to the example given above. It is therefore adequate to use the accuracy of the classifier instead of f-measure, which is calculated using equation 5-9.

$$Accuracy = \frac{Correct\ Classified\ Instances}{Total\ Instances} * 100 \qquad (5\text{-}9)$$

5.4.1 Statistical Analysis of Accuracy

This section reports on the impact of the variations of four parameters SF, SM, WS, and CM (input) on classification accuracy (output). The tables and figures presented in this section are direct outputs of the analysis carried out using the statistical software SPSS.

In a preliminary analysis, it was found that one of the investigated classifiers (KStar) showed a dominant behaviour in form of strong sensitivity to changes in the input parameters (SF, SM, and WS). This led to the conclusion that the initial results of the ANOVA were not highlighting the correct influence of the parameter changes and it was therefore decided to exclude KStar for the main analysis in order to avoid a misinterpretation of the overall impact of parameters on CA.

5.4.1.1 Data Set Bao et al.

The ANOVA results (presented in Table 5–2), showed that 49% of the variations in the dependent variable (classification accuracy) are described by the four input parameters. This means that other input parameters that were not tested in the scope of this experiment may have further influence on the accuracy. Such a result is not surprising, as the investigated problem is highly complex and it is understandable that factors such as the test subject itself and the recorded movement have also an impact on the resulting accuracy.

Table 5–2: ANOVA output for the CA as the dependent variable (Bao et al. data set)

Tests of Between-Subjects Effects

Dependent Variable: CA

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	3257844[a]	670	4862	260	.000
Intercept	1503713645	1	1503713645	80275936	.000
SF	116380	5	23276	1243	.000
WS	216554	31	6986	373	.000
SM	650201	5	130040	6942	.000
CM	1961904	7	280272	14962	.000
SF * SM	5924	25	237	13	.000
SF * CM	7591	35	217	12	.000
SF * WS	36024	155	232	12	.000
SM * WS	60930	155	393	21	.000
WS * CM	92164	217	425	23	.000
SM * CM	110091	35	3145	168	.000
Error	3439779	183633	19		
Total	1510410864	184304			
Corrected Total	6697622	184303			

a. R Squared = .486 (Adjusted R Squared = .485)

Table 5–2 shows the *Sums of Squares* (a measure for the average variability in the data), *Degree of Freedom* (*df* – scores that are free to vary once the mean of the set of scores is known), *Mean Square* (used to estimate the variance), *F* (*F-Ratio* represents the indicator for the significance on performance caused by the independent variables instead of chance), and *Sig* (indicating the significance level at which the main/two-way interaction effects are significant (*Sig.* < 0.05) or non-significant (*Sig*. > 0.05)) for all main and two-way interaction effects. The important columns in Table 5–2 are the Source and Sig. columns as they highlight which main and two-way interaction effects are important in influencing the classification accuracy.

Table 5–2 shows that the four main (SF, WS, SM, CM) and the six two-way interaction effects (SM and CM, WS and CM, SM and WS, SF and WS, SF and CM, SF and SM) have a significant impact on the accuracy. Furthermore, the Type III Sum of Squares can be used as an indication for the importance of the main and two-way interaction effects. It is observed that CM is the most influential factor followed by SM, WS, and SF in decreasing order. The significance for the two-way interaction effects starts with SM and CM and is followed in decreasing order by WS and CM, SM and WS, SF and WS, SF and CM and SF and SM.

While Table 5–2 is best used to decide which parameters are significant for the dependent variable (here classification accuracy), the graphs in this section illustrate the significant two-way interaction effects. An interaction effect outlines a changing main effect of one factor for different levels of a second factor and is therefore of higher interest than the main effect alone, if it is identified as being significant. The individual graphs will illustrate how the classification accuracy is influenced by two of the four different parameters over their respective range of parameter values.

Figure 5–5 shows the interaction effect of changes in the segmentation method on the different classifier methods. The Naïve Bayes classifier shows only a minor improvement in the classification accuracy compared to the remaining seven classifiers that show a substantial classification accuracy increase for an increased segmentation overlap using the FOSW segmentation method. Another observation is the good classification accuracy of the different classifier methods when using the SWAB segmentation method. The classification accuracy is mostly better compared to the result using FOSW with 75% overlap. For Naïve Bayes, the classification accuracy with SWAB decreases below the classification accuracy results achieved with the FNSW segmentation method.

Figure 5–6 presents the effect of an increased window size on the different classifier methods. Naïve Bayes is the only classifier method for which an increased window size directly translates to an increase in classification accuracy. The other classifier methods show an increase in classification accuracy until the window size reaches 7 s, after which it stagnates and starts to decrease for window sizes above 9 s.

Figure 5–7 illustrates the two-way interaction effect between window size and segmentation method. In the range of 0.5 s to 9 s an increase in window size, results in an increased classification accuracy for each tested segmentation method. Window sizes above 9 s result in a changed behaviour. Longer window sizes decrease the classification accuracy for each segmentation method. While the general behaviour is similar for the different segmentation methods, the change in classification accuracy reduces with an increased segmentation overlap, showing a less significant impact on FOSW with 90% compared to FOSW with 25% overlap. The SWAB segmentation method follows the behaviour of FOSW with 75% overlap with a decreased overall classification accuracy. The figure also illustrates that a window size of 6.5 s to 11 s results in the best accuracy.

Figure 5–8 shows the interaction effect between window size and sampling frequency.

An increased window size increases the classification accuracy for all six sampling frequency levels until the window size reaches 9 s. After this point the classification accuracy decreases. For a sampling frequency of 10 Hz, the slope of decrease is flatter compared to the other sampling levels. The graph also highlights that sampling frequencies above 10 Hz result in the best classification accuracy for shorter window sizes, while the 10 Hz sampling level requires a 2 s larger window size.

The last effect under investigation is SF and CM. The graph is not presented, as there is no interaction effect between the different CM levels. The only effect that exists is a minor improvement of accuracy for a change of sampling frequency from 10 Hz to 20 Hz with a nearly constant accuracy thereafter for all CM. This corresponds with the statement in [44] that a sampling frequency above 20 Hz has only a marginally effect on the classification accuracy.

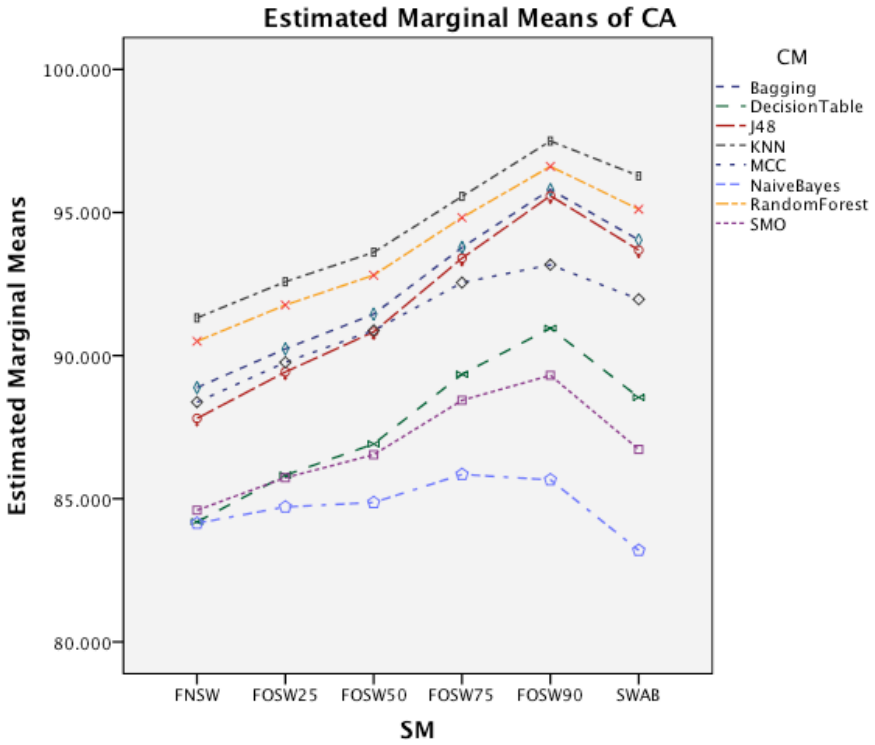


Figure 5–5: Two-way interaction effect for SM and CM

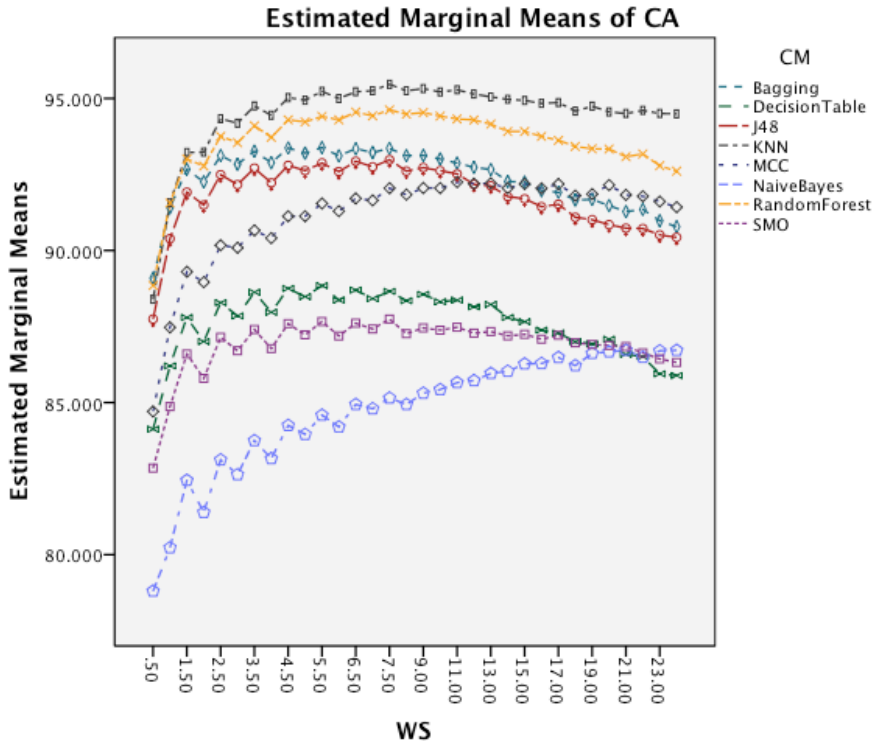


Figure 5–6: Two-way interaction effect for WS and CM

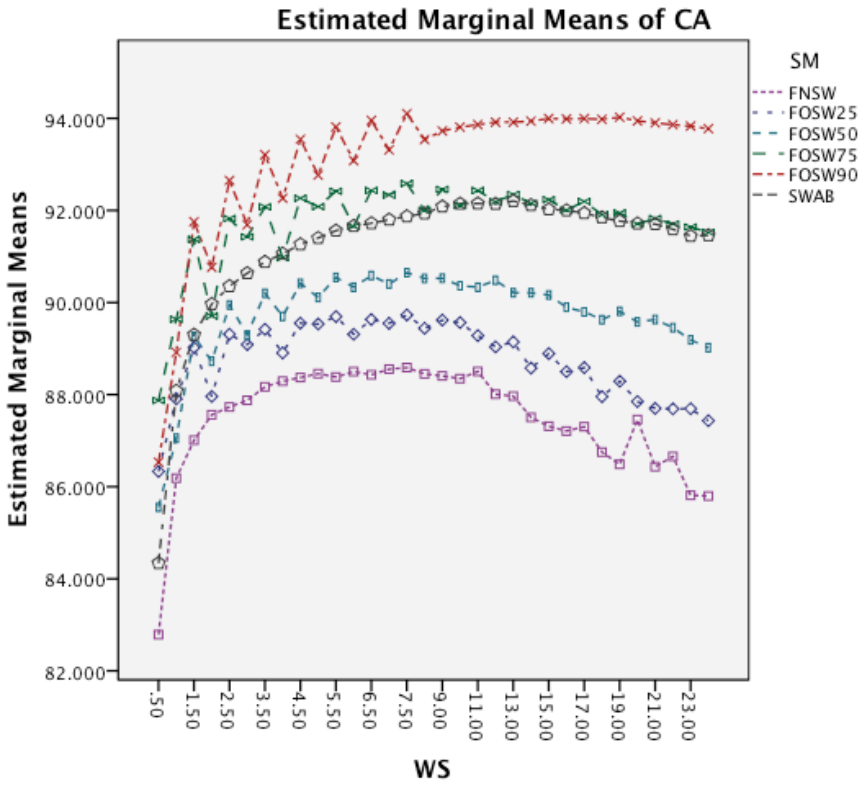


Figure 5–7: Two-way interaction effect for WS and SM

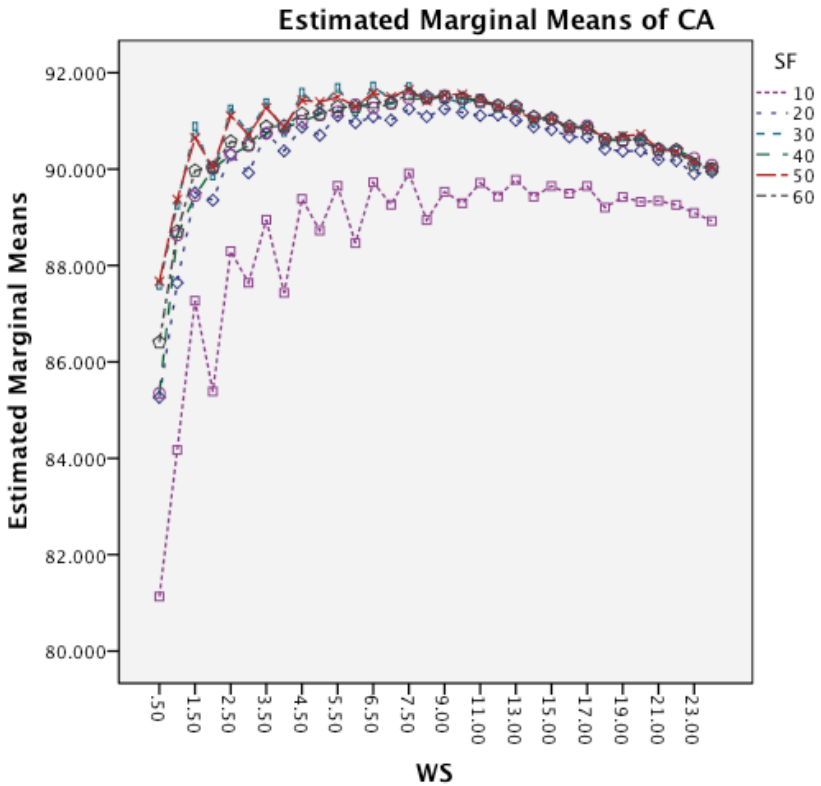


Figure 5–8: Two-way interaction effect for WS and SF

5.4.1.2 Data Set Opportunity

The ANOVA results (presented in Table 5–3), excluding KStar for the same reason that is mentioned in Section 5.4.1, showed that 54% of the variations in the dependent variable (CA) are described by the four input parameters. As mentioned in Section 5.4.1.1, other input parameters that were not tested in the scope of this experiment may have further influenced CA.

Table 5–3: ANOVA output for the CA as the dependent variable (Opportunity data set)

Tests of Between-Subjects Effects

Dependent Variable: CA

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	1785658[a]	670	2665	45	.000
Intercept	147805613	1	147805613	2587861	.000
SF	9501	5	1900	33	.000
WS	183628	31	5923	104	.000
SM	639572	5	127914	2240	.000
CM	759536	7	108505	1900	.000
SF * SM	778	25	31	.545	.968
SF * CM	1445	35	41	.723	.886
SF * WS	4898	155	32	.553	1.000
WS * CM	25038	217	115	2	.000
SM * CM	70397	35	2011	35	.000
SM * WS	90865	155	586	10	.000
Error	1540791	26977	57		
Total	151132062	27648			
Corrected Total	3326449	27647			

a. R Squared = .537 (Adjusted R Squared = .525)

Compared to the table earlier, the Sig. column shows this time, that the three two-way interaction effects containing SF (SF and SM, SF and CM, SF and WS) are non-significant for this data set. The data is also sorted based on the Type III Sum of Squares for an easier overview of the importance of the main and two-way interaction effects. The influential factors CM, SM, WS, and SF have the same decreasing order as the data set provided by Bao et al. The order of importance of the different two-way interaction effects changed and the most significant parameter, based on the Sum of Squares, is now SM and WS, followed in decreasing order by SM and CM, and WS and CM with the rest being non-significant.

Figure 5–9 illustrates the interaction between window size and segmentation method. It is noticeable that an increased window size results in a decrease in classification accuracy for each segmentation method. While the SWAB method

shows a near linear decrease in accuracy, the FOSW and FNSW segmentation methods show a non-linear variation with significant changes in classification accuracy for longer window sizes.

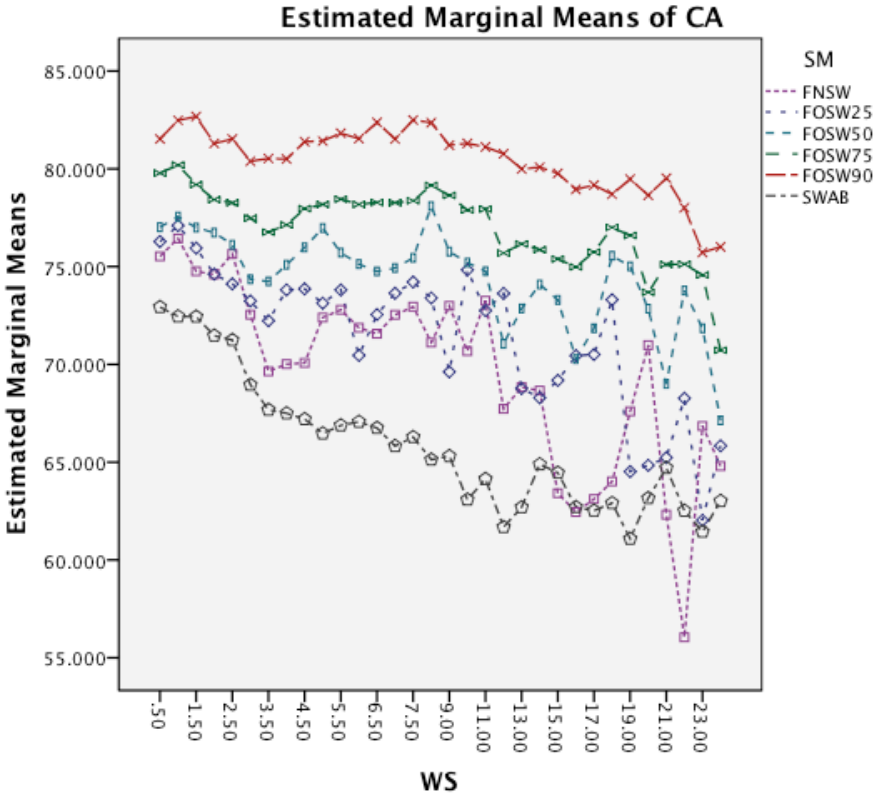


Figure 5–9: Two-way interaction effect for WS and SM

Figure 5–10 shows the interaction effect of changes in the segmentation method on the classifier method. An increased overlap for the FOSW segmentation method results only in minor improvement in classification accuracy for the Naïve Bayes classifier, while it allows for substantial improvements in the remaining seven classifiers. Another observation is the poor performance achieved with the SWAB segmentation method. In the Bao et al. data set, SWAB outperformed classification accuracy results generated using the FOSW segmentation method with 75% overlap, while in the Opportunity data set, the accuracy classification results are the lowest achievable.

Next the interaction effect between window size and classification method is illustrated in Figure 5–11. All classifier show a decrease in classification accuracy for longer window sizes. The effect is less significant on Naïve Bayes compared to the other classifiers. In general, the ANOVA for the second data set highlights better classification results for shorter window sizes.

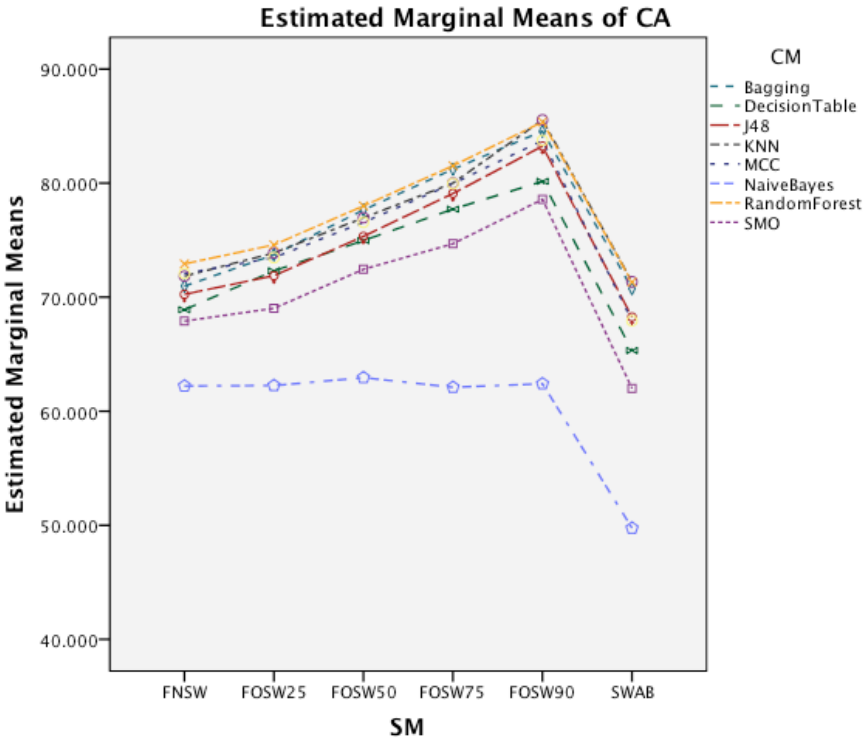


Figure 5–10: Two-way interaction effect for SM and CM

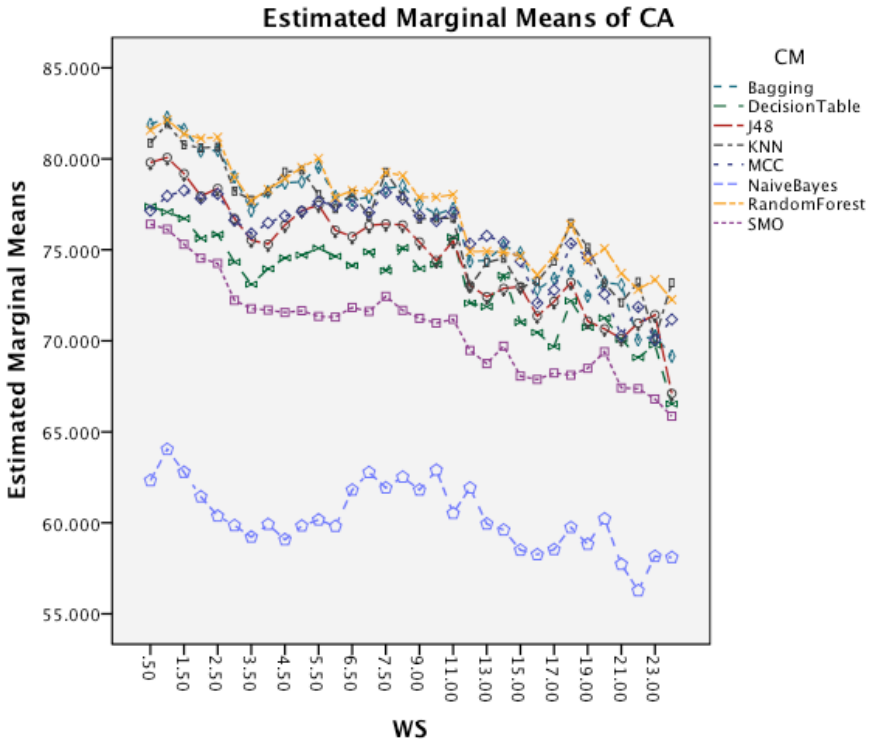


Figure 5–11: Two-way interaction effect for WS and CM

5.4.2 Statistical Analysis of Computational Load

The selection of different parameter values for SF, SM, WS, and CM does not only have an impact on the classification accuracy but also on the Computational Load (CL) of the system. The CL for the classification of ADL events is based on two main factors. The first one is the data pre-processing and feature extraction step (indicated as $Stage_{Time1}$ in Figure 5–12) and the second factor is the actual event classification (indicated as $Stage_{Time2}$ in Figure 5–12). $Stage_{Time1}$ depends on the selected SF, SM, and WS value, excluding any other pre-processing steps such as filtering, which is not of interest in this work, while $Stage_{Time2}$ purely depends on the selected CM. For real-time applications, the combination of SM and WS introduces a limitation for certain parameter combinations, leading to the requirement that equation 5-10 needs to be fulfilled.

$$Stage_{Time1} + Stage_{Time2} \leq Max\ CL \tag{5-10}$$

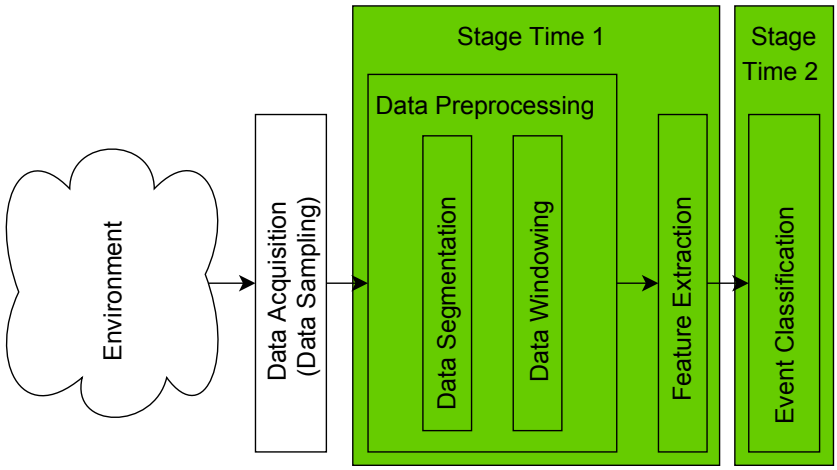


Figure 5–12: Timing factor for computational load

The author therefore conducted an analysis with the CL as the dependent variable to investigate the influence of the four input parameter SF, SM, WS, and CM. In the preliminary analysis, one of the levels of the SM input showed to have a high influence on the dependent variable. As before with KStar superimposing on parameters for the accuracy, the SWAB segmentation method increases noticeably the CL as compared to the other methods. Therefore, the analysis will outline the overall input variables without the SWAB segmentation method.

5.4.2.1 Data Set Bao et al.

The result of the ANOVA is represented in Table 5–4, outlining that 48% of the variation in the dependent variable CL are described by the variation in the input parameters. The source column is sorted based on the Sum of Squares to allow for an easier recognition of the importance of an input parameter. The data highlights that the most important factor for the CL is CM (with SWAB included, this effect was actually non-significant). This is followed by WS, SF, and as the least significant parameter SM. For the two-way interaction effect the five significant combinations are WS and CM, SM and CM, SF and WS, SM and WS followed by SF and SM. The two-way interaction effect between SF and CM is non-significant (Sig > 0.05).

Table 5–4: ANOVA output for the CL as the dependent variable (Bao et al. data set)

Tests of Between-Subjects Effects

Dependent Variable: CL

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	1.316[a]	626	.002	222	.000
Intercept	7.755	1	7.755	818571	.000
SM	.013	4	.003	355	.000
SF	.104	5	.021	2202	.000
WS	.330	31	.011	1122	.000
CM	.515	7	.074	7769	.000
SF * CM	.000	35	3.587E-006	.379	1.000
SF * SM	.001	20	5.456E-005	5.759	.000
SM * WS	.034	124	.000	29	.000
SF * WS	.082	155	.001	56	.000
SM * CM	.103	28	.004	390	.000
WS * CM	.134	217	.001	65	.000
Error	1.449	152957	9.474E-006		
Total	10.521	153584			
Corrected Total	2.766	153583			

a. R Squared = .476 (Adjusted R Squared = .474)

Figure 5–13 illustrates the interaction effect between window size and classifier method. All classifiers require a longer computational load for longer window sizes. Moreover, a small drop in the computational load is noticeable for short window sizes about 2 s. The graph also presents that the rate in which the computational load increases is higher for MCC and SMO.

The two-way interaction effect of segmentation method and classifier method illustrated in Figure 5–14 shows that there is a significant improvement in computational load for MCC and SMO for an increased overlap. The decrease of computational load for SMO is thereby steeper than compared to MCC. The

remaining classifiers show only minor changes in computational load for an increased window overlap.

Figure 5–15 illustrates the interaction effects between window size and sampling frequency. The longer window size results in a higher computational load for all sampling levels. Another observation is that a higher sampling frequency increases the effect of longer window sizes.

The last interaction effect under investigation is window size and segmentation method. The graph in Figure 5–16 shows that this interaction effect has similar patterns to the other two interaction effects containing the window size (WS and CM, WS and SF). All segmentation methods have a significant increase in computation load for a longer window size. An interesting observation is that the segmentation methods with higher overlap result in a lower computational load for longer window sizes.

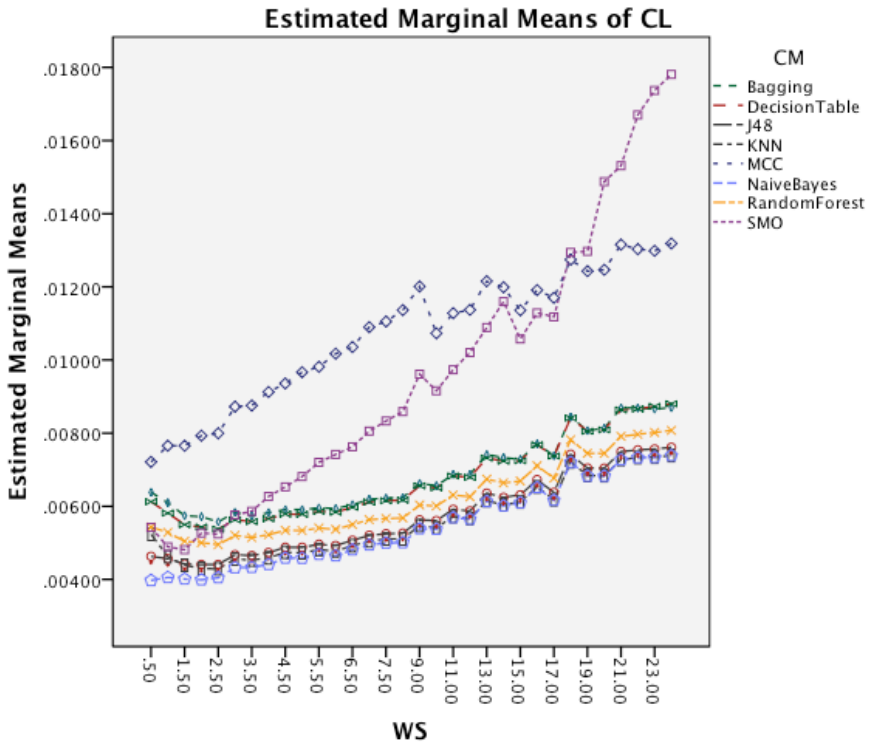


Figure 5–13: Two-way interaction effect for WS and CM

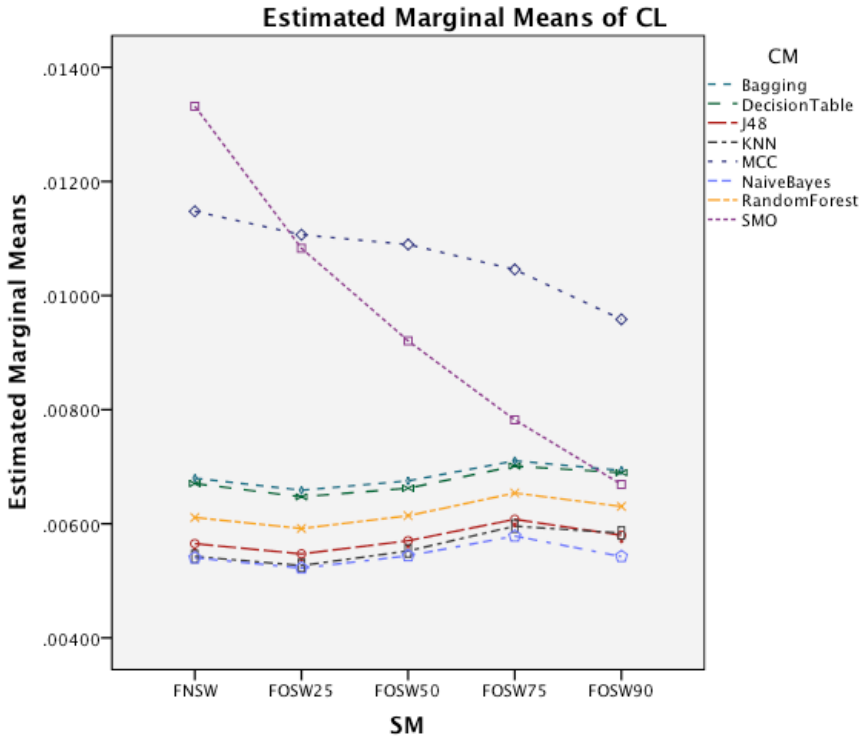


Figure 5–14: Two-way interaction effect for SM and CM

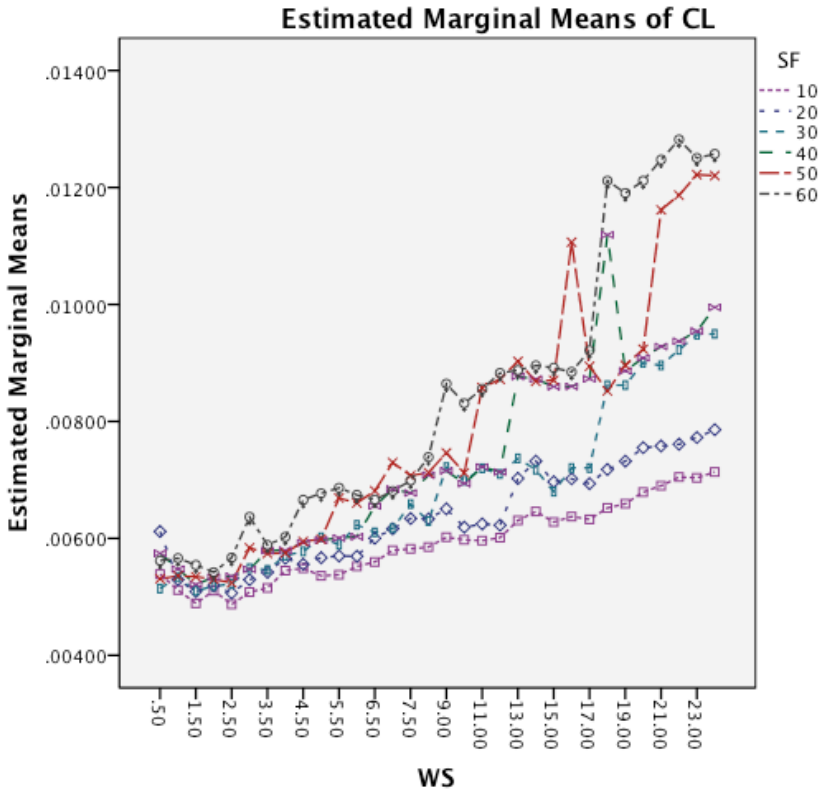


Figure 5–15: Two-way interaction effect for WS and SF

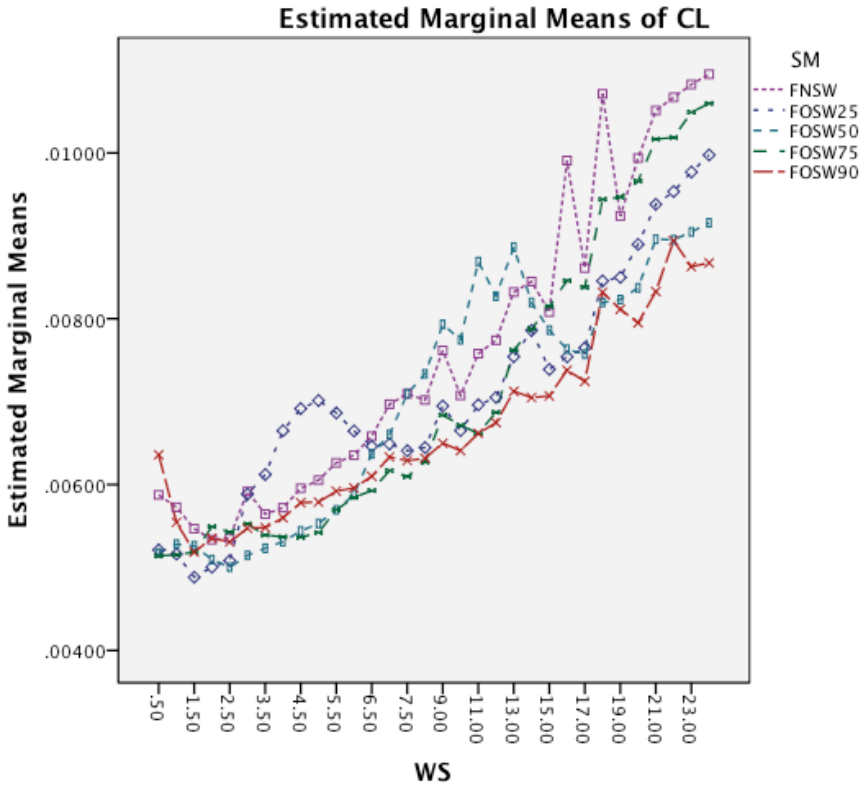


Figure 5–16: Two-way interaction effect for WS and SM

5.4.2.2 Data Set Opportunity

The result of the ANOVA is represented in Table 5–5, outlining that 72% of the variation in the dependent variable CL are described by the variation in the input parameters. This is a nearly 30% increase from the data set provided by Bao et al. The source column is sorted based on the Sum of Squares to allow for an easier recognition of the importance of an input parameter. The data highlights that the most important factor for the CL is CM. This is followed by WS, SF and as the least significant parameter SM, which is the same outcome of the Bao et al. data set. For the two-way interaction effect, the four significant combinations are WS and CM, SM and CM, SF and WS, followed by SM and WS. Two of the three interaction effects containing SF (SF and SM, SF and CM) are non-significant. This is a change from the first data set where only SF and SM was non-significant.

Table 5–5: ANOVA output for the CL as the dependent variable (Opportunity data set)

Tests of Between-Subjects Effects

Dependent Variable: CL

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	.358[a]	626	.001	93	.000
Intercept	1.525	1	1.525	247087	.000
SM	.008	4	.002	328	.000
SF	.021	5	.004	671	.000
WS	.090	31	.003	470	.000
CM	.112	7	.016	2595	.000
SF * SM	.000	20	7.884E-006	1	.181
SF * CM	.000	35	7.236E-006	1	.223
SM * WS	.006	124	4.866E-005	8	.000
SF * WS	.011	155	6.966E-005	11	.000
SM * CM	.050	28	.002	290	.000
WS * CM	.060	217	.000	45	.000
Error	.138	22413	6.172E-006		
Total	2.021	23040			
Corrected Total	.496	23039			

a. R Squared = .721 (Adjusted R Squared = .714)

Figure 5–17 shows the interaction effect between window size and classifier method. For longer window sizes, the computational load increases at various rates for all classifiers. The highest increase is observable for SMO. The slope with that the computational load increases for MCC is lower in this data set compared to the earlier one. Instead of showing a similar behaviour to SMO, it behaves more like the remaining classifier methods.

Figure 5–18 illustrates the interaction effect between segmentation and classifier method. Only SMO shows significant improvements with regard to the

computational load for an increased segmentation overlap. In comparison, the remaining classifiers show only minor changes.

Figure 5–19 shows the interaction effects between window sizes and sampling frequencies. For an increase in window size, the computational load increases for all sampling frequency levels. The rate of increase is linked to the sampling frequency, where a higher frequency means a higher rate in the increase of computational load.

The last interaction effect under investigation is window size and segmentation method. Figure 5–20 illustrates similar behaviour to the interaction effect of window size and classifier method. All segmentation methods have a significant increase in computational load for longer window sizes. Both data sets investigated, show that the effect an increased window size has, reduces for an increased overlap, hence flatter increase rates for the computational load.

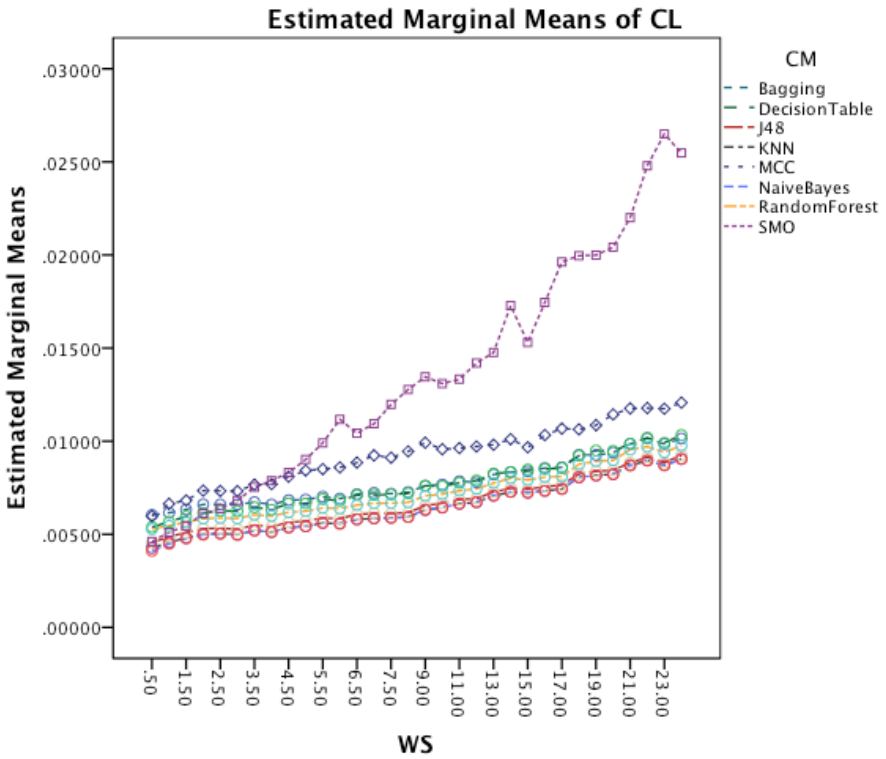


Figure 5–17: Two-way interaction effect for WS and CM

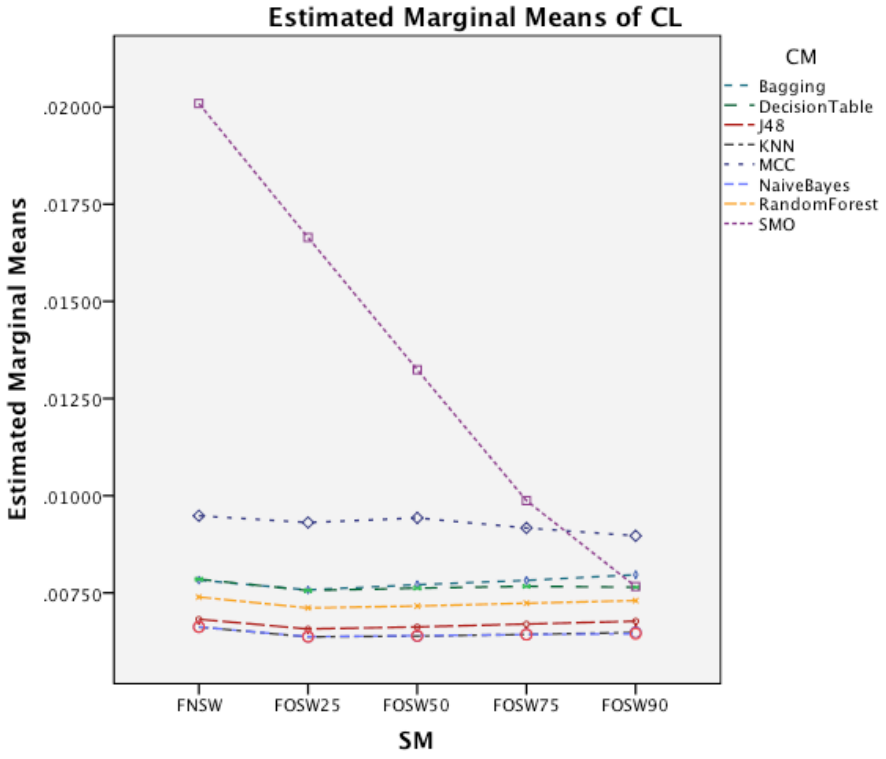


Figure 5–18: Two-way interaction effect for SM and CM

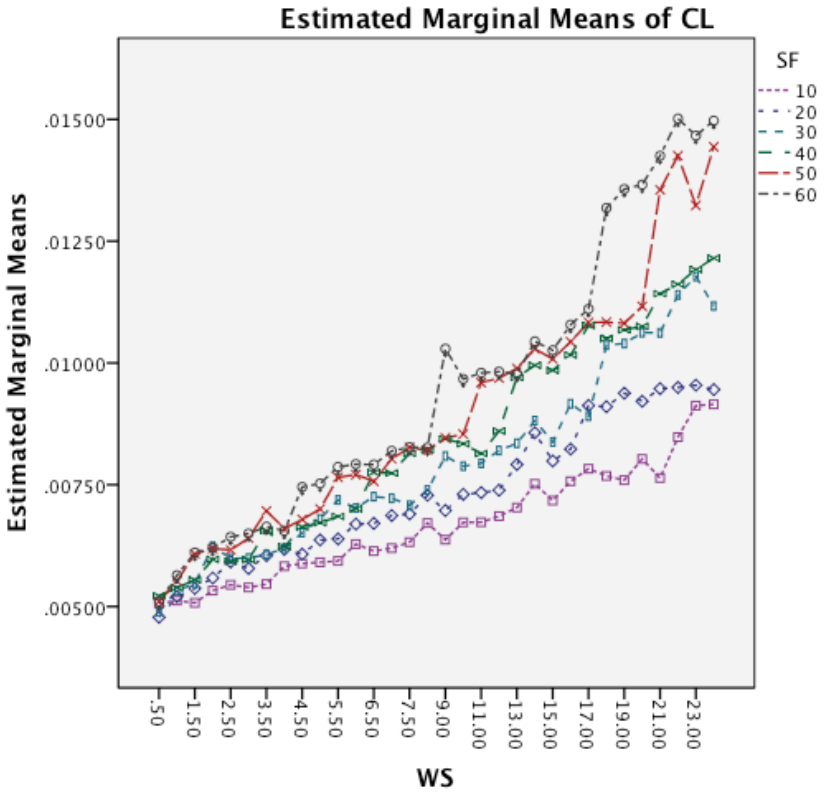


Figure 5–19: Two-way interaction effect for WS and SF

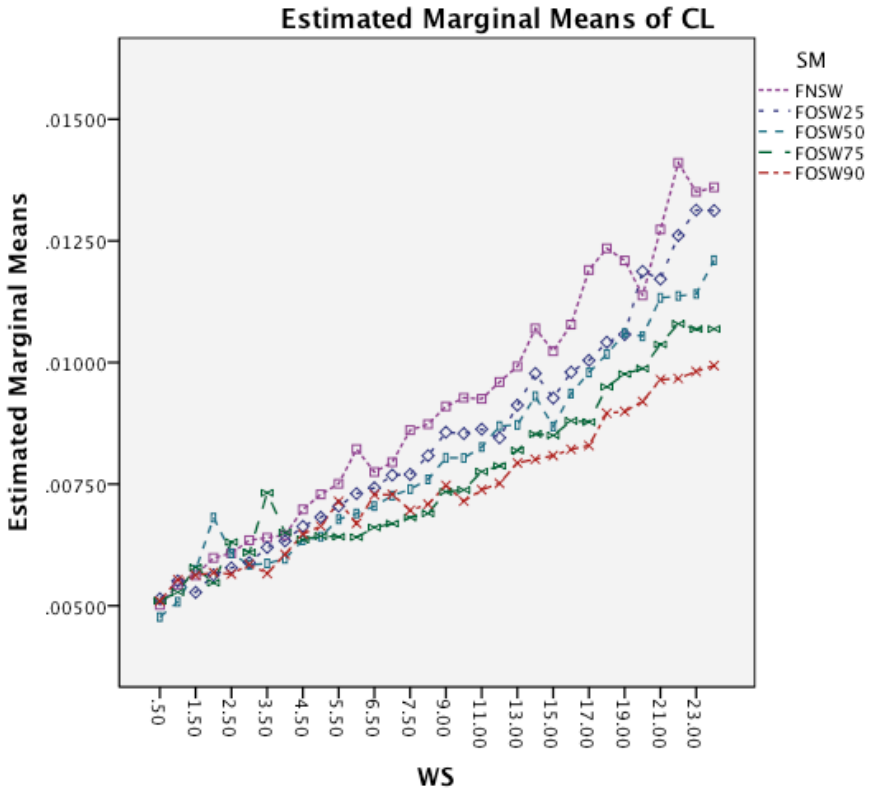


Figure 5–20: Two-way interaction effect for WS and SM

5.4.3 Parameter Selection

The Analysis Of VAriance described in Section 5.4.1 and 5.4.2 using different parameters and their combinations increase the understanding of their influence with regards to ADL event Classification Accuracy (CA) and Computational Load (CL). However the inevitable question still stands: what is the best combination of parameters for a given requirement? Here the answer depends strongly on the preference with respect to classification performance, e.g. is the best CA required or are there limitations to CL, such as highlighted in equation 5-10. Therefore, a set of well-performing parameter sets based on the trade-off between CA and CL were identified. For a given data set certain parameter combinations will achieve a similar CA but will require different CL and vice versa. When plotted in a graph, such as presented in Figure 5–21, the best CA for a given CL would follow the black line (called Pareto frontier), with dominated parameter sets lying on the left hand side of the curve. Hence, a parameter set is dominated if there is a combination of parameter values that may result in a similar level of CA with less CL or achieve better CA with the same CL. The Pareto frontier, also referred to as Pareto curve, outlines the set of non-dominated (also called dominating) solutions, herein represented by a set of parameter combinations. One set of parameter values may achieve best CA at the cost of a high CL (Point 1) and another combination will achieve the lowest CL at the cost of a lower accuracy (Point 2). Parameter sets in between points 1 and 2 on the Pareto frontier are subject to a trade-off (Point 3), hence accepting the sacrifice of either, accuracy or CL, depending on the context of the applications or potential corresponding limitations, e.g. hardware constraints.

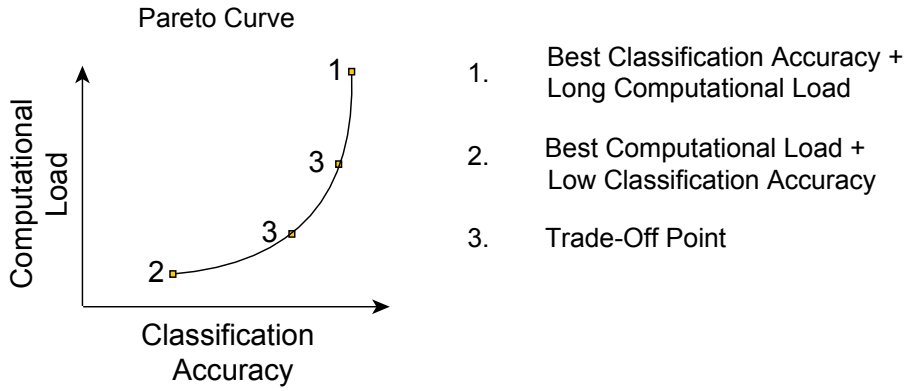


Figure 5–21: Explanation of the Pareto curve

5.4.3.1 Data Set Bao et al.

Figure 5–22 illustrates two separate Pareto curves for the Bao et al. data set. The illustration shows a non-limited Pareto curve over all 10368 possible parameter combinations and a Pareto curve that is limited to a 10 Hz sampling frequency, as this is the hardware limitation for the OHDAD platform presented in Chapter 3. The table containing the dominating parameter combinations can be found in Appendix G. For the non-limited Pareto curve, with 10368 possible combinations, only 20 are dominating.

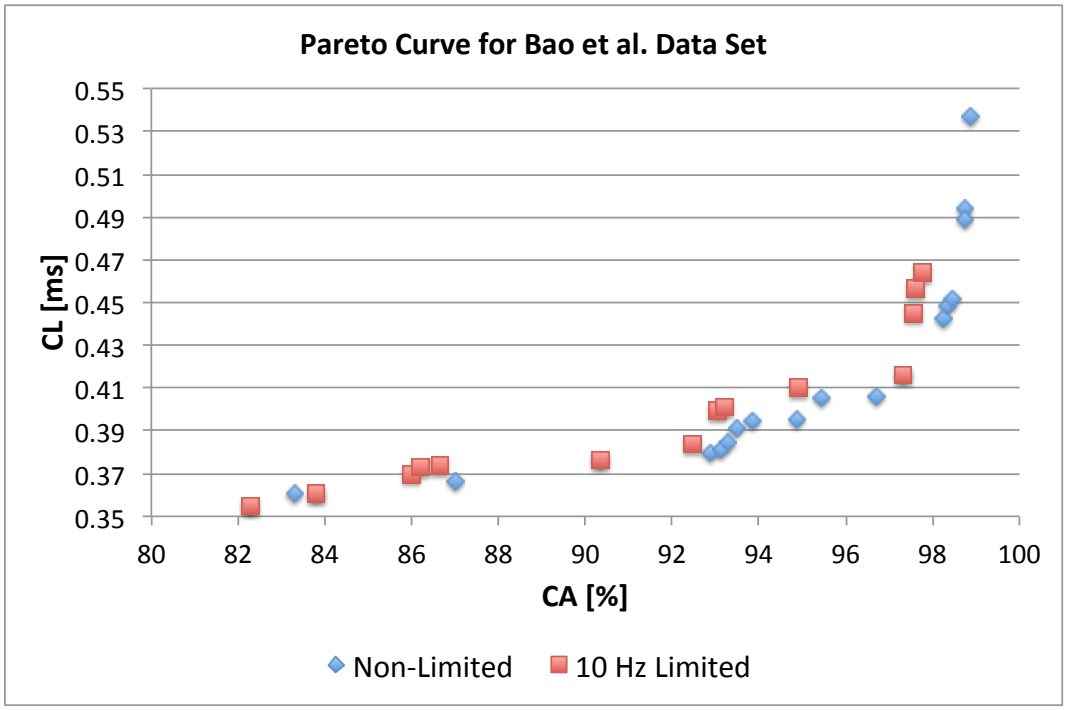


Figure 5–22: Dominant points on the Pareto curve for the Bao et al. data set

The parameter combinations show a maximum SF of 30 Hz with equal occurrence of 20 Hz and 30 Hz (8 combinations each). For the SM parameter only the FOSW method with various overlaps percentages shows to be of importance. An overlap value of 90% shows higher CA, while an overlap value of 25% results in shorter CL. The dominating points highlight short window sizes of 1.5 s for the shortest CL and a WS of 7.5 s for best CA. The dominating classifiers are Naïve Bayes and KNN. The latter is in the majority of combinations and results in the best CA, while the former results in the shortest CL. The achievable CA ranges from 82.2% with a CL of 0.355 ms to 98.8% with a CL of 0.537 ms. The 10 Hz SF limited parameter combinations highlight only 14 dominating points on the Pareto curve. The SM shows similar behaviour to the non-limited data set; with a higher influence of the FOSW method with 75% overlap. The WS is around 1.5 s for shortest CL and around 7.5 s for best CA. The CM parameter for the 10 Hz limited case is

nearly identical to the non-limited case; with the exception of the presence of J48 as an influential algorithm in in the former case. The achievable CA ranges from 82.2% with a CL of 0.355 ms to 97.7% with a CL of 0.464 ms.

5.4.3.2 Data Set Opportunity

Figure 5–23 illustrates the two separate Pareto curves for the Opportunity data set, based on a non-limited and a 10 Hz limited parameter combination set. The table containing the dominating parameter combinations can be found in Appendix H. For the non-limited Pareto curve, 12 dominating points are identified, while the 10 Hz limited data set shows only 6 dominating points.

The full data set shows SF parameter values in the range of 10 Hz to 30 Hz with a majority split between 10 Hz and 20 Hz. FOSW with 90% overlap (SM) dominates for higher CA and with 25% overlap for lower CL in both data sets. Short WS have a significant influence in this data set, as both data sets include parameter combinations with 1.5 s or less. The dominating classifiers are Naïve Bayes, and KNN. This is a slight variation compared to the 10 Hz limited Bao et al. data set. The non-limited data set has a range of 64.2% to 94.3% in CA with a CL of 0.376 ms to 0.443 ms. The 10 Hz limited data set ranges from 63.2% to 92.2% for CA with 0.383 ms to 0.43 ms for CL.

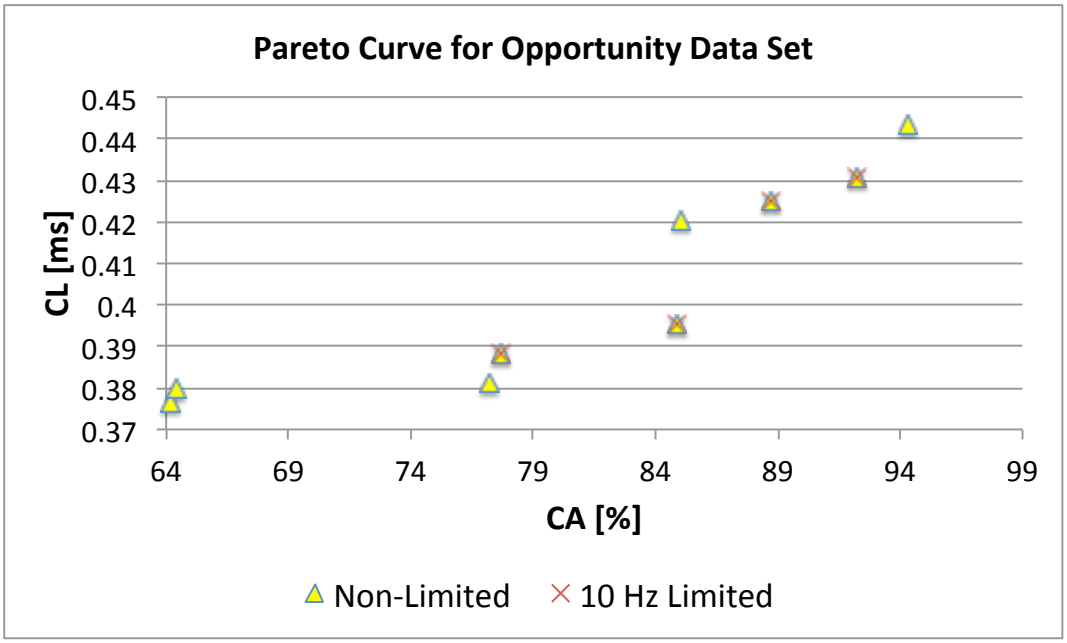


Figure 5–23: Dominant points on the Pareto curve for the Opportunity data set

5.5 Summary and Discussion of Results

The work carried out as part of this chapter investigated the influence of a set of different parameter combinations on the ADL event Classification Accuracy (CA) and Computational Load (CL) of a system. Using accelerometer data from two independent open access data sets, the author was able to analyse individual data of 23 different test subjects. Even tough in [87], the authors argue that the use of any one of two activity classification methods, uniform (where the training data comes from all tests subjects) and individual (training data representing separate test subjects) can lead to problems; namely, generalization (arising from the uniform method) and a small training data set (individual), which can both result in poor performance. The research and associated experiments presented here fall in the individual category as performance measures (CA and CL) are generated for each of the 23 test subjects involved. The author believes that despite the pitfalls described above, this was the better method to adopt; this is also in line with Elbert et al.'s approach. Furthermore, one of the performance parameter was chosen to be classification accuracy, with other possible measures being precision, recall, and f-measure. However, as highlighted in [52] the other parameters are best used for an imbalanced data set, e.g. when normal and abnormal data classes are of interest. As the tested activity set has 6 and 4 class labels, which are nearly equally represented and important to classify, the accuracy measure can be used without loss of validity.

In summary the outputs of the work presented here, are listed below:

- The importance of parameters for CA ranked in order of decreasing influence is CM, SM, WS and SF
- The impact of WS is different for both data sets
- Increased segmentation overlap improves CA
- The influence of SWAB on CA is different in both data sets
- SF above 10 Hz has only a minor improvement on CA
- CL behaves the same for both data set
- The importance of parameters for CL ranked in order of decreasing influence is CM, WS, SF, and SM
- Some dominant parameter (CM, SM, SF) combinations of the Pareto curve are similar for both data sets
- Higher CL does not automatically result in higher CA

The following discussion will look into the results of the ANalysis Of VAriance (ANOVA) for CA and CL and finish with the dominant parameter points of the Pareto curves. The two-way interaction effect between SM and CM highlights for both data sets that FOSW with 90% overlap results in the best CA. From FNSW (no overlap) to FOSW with 90% overlap, both data sets show that more overlap improves the CA. A possible reason for this is that the increase in overlap allows for a bigger training set and has the lowest loss of information, in the range of investigated SM. The results for SWAB are mixed. For the Bao et al. data set CA is just below CA for FOSW with 90% overlap, while the Opportunity data set showed SWAB to be the worst SM tested. Further research needs to look into the actual benefit of a dynamically sliding window, which incidentally was reported in [52] as giving good results, as the results reported here (in terms of CA) are inconsistent between the two data sets. Another difference between the data sets is observed for the WS and CM two-way interaction effect. While for the first data set (Bao et al.), the CA improves for WS between 1 s and 8 s and only decrease for WS values above 8 s, the second data set (Opportunity) achieves best CA for 0.5 s and starts to decrease immediately after that. A similar behaviour can be seen for the two-way interaction effect of WS and SM with both data sets (compare Figure 5–6 and Figure 5–7 in Section 5.4.1.1 and Figure 5–9 and Figure 5–11 in Section 5.4.1.2). Researchers should therefore choose smaller WS if possible. Another difference between the two data sets is the significance of the two-way interaction effect WS and SF; namely, significant interaction (but not the most significant) for Bao et al. and non-significant interaction for the Opportunity data set. The graph in Figure 5–8 (see Section 5.4.1.1) shows that sampling frequencies above 10 Hz achieve nearly the same CA, while the 10 Hz sampling frequency is marginally lower, endorsing the finding in [44] that sampling frequencies above 20 Hz result in only minor accuracy gain.

For CL, both data sets show the same behaviour for the two-way interaction effects. For the three interaction effects including WS (WS and CM, WS and SM, WS and SF) similar behaviour is observable. A shorter WS results in a lower CL, while a longer WS will increase the CL. This effect is lowest for WS and CM and highest for WS and SM. The interaction effect between SM and CM highlights no significant change for any classifier besides SMO. SMO is the only classifier that can reduce the CL with an increased segmentation overlap.

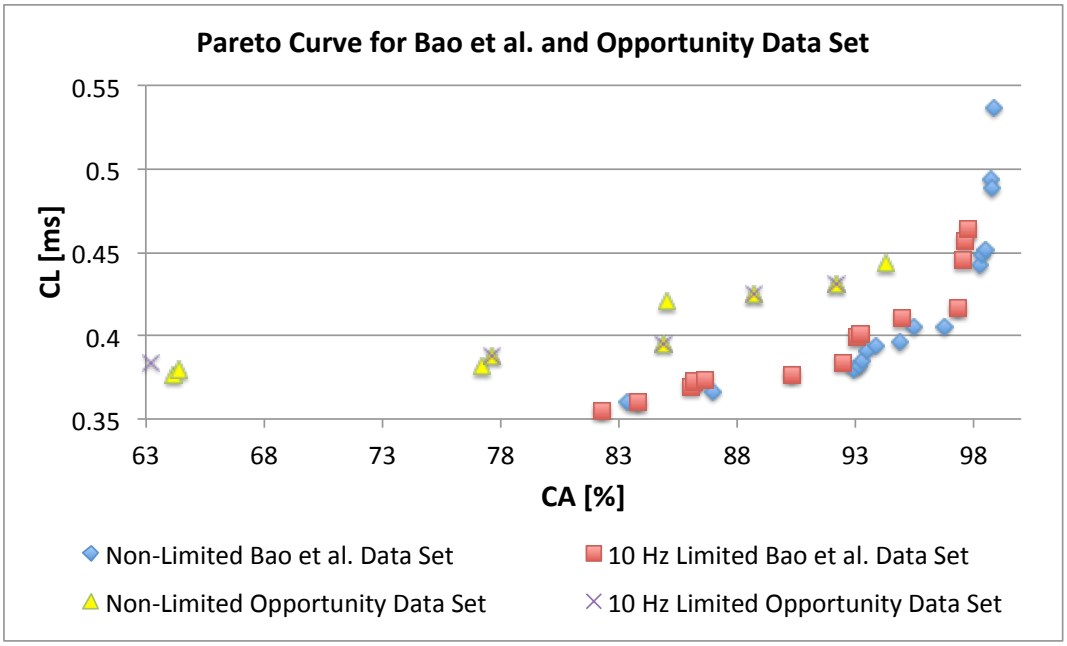


Figure 5–24: Dominant points on the Pareto curve for both data sets

The author used ANOVA to quantify the influence of the different parameters SF, SM, WS, and CM on CA and CL. In addition, a Pareto curve based approach is used to highlight dominant parameter combinations for "optimum" achievable performance (optimality being decided by the user in a given context/application). Figure 5–24 presents the four earlier introduced Pareto curves based on the dominant combinations. The illustration shows that all graphs have a similar outline and that it is possible to achieve similar results irrespective of the data set. This is highlighted with only a 4.3% difference in CA between the two top performing parameter combinations. However, the dominant parameter combinations are different for each data set. Therefore, it is not possible to present a single combination that will work best for all data sets.

Having said that, some dominant points have similar parameter combinations. In both data sets high CA is achieved with the KNN classifier and the FOSW with 90% overlap for SM. Furthermore, the Pareto points show that a sampling frequency above 30 Hz is not necessary and only minor improvements in CA are achieved with a SF above 10 Hz. As a consequence, it is recommended to adjust parameters individually for each data set and test subject to achieve optimal results, especially with regards to WS. The Pareto curves also reveal that a higher CL does not necessarily result in better CA, as the algorithms under investigation are not recursive. The Pareto curve is also a good tool to investigate the influence of a hardware limitation such as a low sampling rate, storage space and battery runtime. When superimposing the hardware limited Pareto curve with the non-

limited curve a simple comparison of achievable CA and CL is possible. For example, data collected with the OHDAD will not achieve the best possible ADL CA but both 10 Hz limited Pareto curves highlighted that the difference in CA will not be significant with the correct parameter selection. The difference between the best limited and non-limited CA for the Bao et al. data set is 1.1% and for the Opportunity data set 2.11%. This is lower than the 5.55% difference between the best CA for the two 10 Hz limited combinations for the two data sets. The results presented in Section 5.4.3 in combination with the ANOVA in Section 5.4.1 and 5.4.2 can be used for future research as a tool to select parameter combinations for AAL event classifications with the sound understanding of how each parameter influences the outcome of event CA and CL.

5.6 Conclusions

This chapter has presented a new instrument to help select data acquisition and processing parameters for the recognition of Activities of Daily Living (ADL). The review of the literature uncovered a lack of consensus in terms of the selection of sampling frequency (SF), segmentation method (SM), and window size (WS) for the recognition of ADLs using different classifier methods (CM). However, to have a fair performance comparison with AIS, good performing parameter combinations need to be selected. Therefore, the impact of the sampling frequency (6 levels), segmentation method (three segmentation algorithms with different parameters resulting in 6 different levels) and segmentation window size (32 levels) on the classification accuracy and computational load of a set of commonly used classifier methods (9 levels) has been investigated. This has involved experimenting with two data sets, containing 20 and 3 test subjects, respectively, and analysing the resulting data using ANalysis Of VAriance (ANOVA). The analysis for the classification accuracy showed that the choice of CM is the most important parameter followed by SM, WS, and finally SF. It also showed that in the case of computational load the parameters ranked in order of decreasing influence are CM, WS, SF, and SM. The results have been presented graphically using a Pareto curve, which highlighted two dominant classifiers for both data sets (KNN, Naïve Bayes). The Pareto curve did not show matching dominant points in both data sets. However, it showed that combinations of three out of the four factors (CM, SM, SF) are likely to result in dominant points. The Pareto curve also helped to identify the expectable loss of classification accuracy by the low sampling rate of 10 Hz by the OHDAD. The investigation showed

only a reduction of around 1% and 2% for the Bao et al. and Opportunity data sets, respectively. The author suggests that the Pareto curve is an effective instrument, which can be used to select sets of parameters based on their impact on classification accuracy and computational load and resolve trade-off issues. In addition, the Pareto curve is used for the parameter selection in the evaluation process in the next chapter.

6 Comparison between Artificial Immune System, Supervised Classifiers, and a Threshold Algorithm

6.1 Introduction

This chapter builds on the work discussed in Chapters 4 and 5 and presents an analysis of the capabilities of detecting abnormal activity in a long-term data stream using three different classification approaches. The three tested approaches are:

- Simple Signal Vector Magnitude (SVM)-based threshold algorithm.
- Unsupervised classifier.
    - Represented by two AIS-based monitoring systems discussed in Chapter 4.
- Supervised classifier.
    - Represented by four different classifier methods discussed in Chapter 5.

The analysis consists of three different scenarios that are generated using the OHDAD and simulation environment that were introduced in Chapter 3. For the analysis the four performance measures precision, recall, f-measure, and computational load are recorded and evaluated. After the motivation for this analysis is presented, the performance measures are introduced, followed by the selection of parameter for the used algorithms. For a better comparison of the different scenarios and the respective results, the former are first described in detail, then the individual results are analysed. In addition, the experimental setup and the results of scenario 1 were also published by the author in a peer-reviewed conference proceeding [111]. The chapter concludes with a discussion and summary. The discussion section will highlight four main outcomes of the analysis in detail. Firstly the simple threshold algorithm is only performing well under specific conditions. Secondly, the AIS algorithm performs well even outside specific conditions. Though the data labelling showed limitation in the AIS framework used. Thirdly, only one of the four tested supervised classifiers outperforms the AIS algorithm. Fourthly, AIS requires a significantly longer computational load.

6.2 Motivation for the Analysis of Different Methods for the Detection of Abnormal Activity

The use of Artificial Immune System (AIS) for the novel approach in detecting abnormal activity in the Ambient Assisted Living (AAL) field can only be recommended when its

performance improves over existing classification methods. To the author's best knowledge no such comparison has been carried out and is therefore the focus of this chapter. In Chapter 4 it was highlighted that the complexity of AIS fits in between simple threshold-based algorithms and more traditional classifiers such as Decision Trees (J48), Artificial Neural Networks (ANN), or K-Nearest Neighbour (KNN). Investigating the detection and false alarm rate of different algorithms gives the author also the opportunity to investigate whether the complexity has any affect on the required Computational Load (CL) for the different algorithms. However, the comparison of CL might not be completely fair, as the supervised classifiers are not implemented using the same programming language (Java vs. Matlab) and were most likely time optimised over the years by the development team of the Weka software package.

6.3 Description of Performance Measure

For the analysis of detection and false alarm rate of abnormal activity, new performance parameters need to be introduced. This is required as only a minor percentage of the generated long-term data set is of interest. The generated data set can be described as being imbalanced. This class of data sets was also mentioned earlier in Chapter 5. The imbalance prevents the use of classification accuracy and instead requires the introduction of precision, recall, and f-measure. The reason for not being able to use accuracy is illustrated by the following case: if the data set has two classes with a distribution of 98% and 2%, guessing only the majority class will result in a 98% classification accuracy [142]. However, for this analysis the 2% minority class (presenting e.g. abnormality) is of interest.

Table 6–1: Confusion matrix of a classifier prediction

		Dataset	
		TRUE	FALSE
Classifier Prediction	TRUE	TruePositive (TP)	FalsePositive (FP)
	FALSE	FalseNegative (FN)	TrueNegative (TN)

Table 6–1 illustrates the four possible outcomes of a classifier prediction: TruePositive (TP - identifying a true data point as true), FalsePositive (FP - identifying a false data point as true), FalseNegative (FN - identifying a true data point as false), TrueNegative (TN - identifying a false data point as false). It should be noted that only TP and TN correctly classify a data point. Furthermore, a data point is associated to the true or false set depending on the application context. In the here described context the abnormal

activity will be a true data point, as the identification of abnormality is of interest. Based on these four states, the new performance measures can be calculated using equation 6-1 (precision), equation 6-2 (recall), and equation 6-3 (f-measure). In terms of detection of abnormality, precision characterises how many of the predicted abnormal data segments in the test set were actually labelled as abnormal. Recall determines how many of the data segments labelled as abnormal in the data set were missed. A value below 1 indicates that the classifier missed at least one data segment. In the worst-case scenario, a missed abnormality (e.g. fall) could have severe health consequences. The f-measure parameter calculates the harmonic mean between precision and recall. Should one of the two parameters be more important than the other, the general equation of f-measure (see equation 6-4) can be used to weight the importance accordingly. For $\beta > 1$ the emphases is on precision, while $\beta < 1$ increases the weight on the recall parameter.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositve} \tag{6-1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{6-2}$$

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{6-3}$$

$$F - Measure_{\beta} = (1 + \beta^2) * \frac{Precision * Recall}{(\beta^2 * Precision) + Recall} \tag{6-4}$$

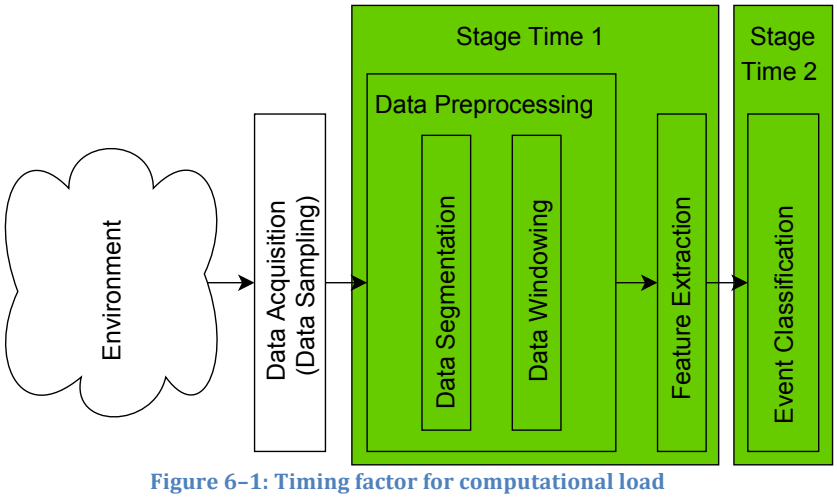


Figure 6–1: Timing factor for computational load

The CL required by the different detection methods is based on two main factors. The first one is the data pre-processing and feature extraction step (indicated as $Stage_{Time1}$ in Figure 5–12) and the second factor is the actual event classification (indicated as

$Stage_{Time2}$ in Figure 5–12). $Stage_{Time1}$ depends on the sampling frequency, segmentation method, window size, and the extracted features, excluding any other pre-processing steps such as filtering which is not of interest in this work. In the work here presented, only the simple threshold and the supervised classifier algorithms require feature extraction, while the AIS algorithms use raw sensor data. The time required for $Stage_{Time2}$ depends purely on the selected algorithm.

6.4 Description of Parameter Selection for the Different Algorithms

After the description of the performance measures, this section discusses the selection of various parameters associated with the algorithms used. A fair comparison between the different approaches for the detection of abnormal activity is only possible when each algorithm is used with parameter combinations that can achieve the best possible classification results. Only in this way is it possible to show the validity and reliability of AIS for AAL applications.

$$SVM = \sqrt{\mathrm{x}^2 + \mathrm{y}^2 + \mathrm{z}^2} \tag{6-5}$$

The threshold-based approach relies upon the Signal Vector Magnitude (SVM) that determines the movement intensity using equation 6-5, with x, y, and z being the acceleration of each axis. This feature was also extracted during the work described in Chapter 5. The required threshold for the SVM parameter was determined by a-priori knowledge of the collected activities and the corresponding sensor data. For each fall activity j (each activity was performed five times) the maximum SVM value of the data segment was determined using equation 6-6, with $i = \{1, 2, \dots, n\}$ being the i-th element of the activity data with length n. From the list of maximum SVM values, the maximum threshold is defined using equation 6-7, with $j = \{1, 2, \dots, 5\}$ indicating the fall repetition. The lowest SVM value recorded for a fall was 2.06 g, with 2.86 g as a maximum. Hence, a slightly lower threshold of 2.04 g is selected.

$$SVM_{FallActivity_j} = \max_i\left(\sqrt{{x_{ji}}^2 + {y_{ji}}^2 + z_{ji}^2}\right) \tag{6-6}$$

$$SVM_{Threshold} = \min_j\left(SVM_{FallActivity_j}\right) \tag{6-7}$$

The limitation of this approach is that the maximum SVM values for the data fragments representing the sitting down activity, fall into the range of 1.62 g to 2.10 g. Lying down showed a similar value range with 1.79 g to 2.07 g. The current threshold of 2.04 g is

going to result in false alarms for sitting and lying down activities with the high SVM value or require missing some of the fall activities in the case of increasing the threshold to 2.10 g. It is therefore no surprise, that more advanced threshold algorithms aimed at falls [81], [82], are based on upper and lower threshold values to reduce possible false alarms.

The parameter values for the supervised classifiers were determined by a-priori knowledge of similar data sets. The parameter combinations presented in Table 6–2 are based on the Pareto curve and the determined dominant points identified as part of the work presented in Chapter 5. For each data set, the parameters for the best classification accuracy with regards to the different dominant classifiers (KNN, J48, and Naïve Bayes) were chosen. In addition, the SMO classifier (based on the Support Vector Machine (SVM) classifier) was selected outside the dominant classifiers, as it was highlighted in [94] for its ability to detect abnormality.

Table 6–2: Used parameter combinations for the supervised classifiers

		Method	Segmentation Method	Window Size [s]
Supervised Classifier	Bao et al., 2004 [25]	KNN	FOSW 90	7.5
		J48		5
		Naïve Bayes		1.5
		SMO		7.5
	Opportunity [23]	KNN		1.5
		Naïve Bayes		0.5

The segmentation method for all classifiers was the FOSW algorithm with 90% overlap. For the window size, one data set (Bao et al.) suggests values of 7.5 s, 5 s, 1.5 s, and 7.5 s for KNN, J48, Naïve Bayes, and SMO, respectively. The other data set (Opportunity) suggests much smaller values of 1.5 s, 0.5 s, and 0.5 s for KNN, Naïve Bayes, and SMO. However, a window size of 0.5 s resulted in insufficient memory problems on the experimental computer used and had to be disregarded. In addition, other parameter combinations that resulted in a lower classification accuracy and therefore lower computational load were ignored, as time is not considered a limiting factor right now. The individual activity segments are labelled using two different methods. The first one keeps the original 13 different activity labels, while the second method labels the fall as abnormality and combines the remaining 12 activities into one normal class. The resulting two-class problem provides a larger training data set for the normal activity.

Table 6–3: Used parameter combinations for the Artificial Immune System

Parameter	Parameter Value
Mature Detector	300
Immature Detector	100
max Affinity Radius	100
min Detector Radius	4
Immature Time	2
Mature Time	4

The optimal parameter selection for the two AIS-based monitoring systems is more complex. In [98] the authors used 100 detectors for their AIS-based intrusion system (LISYS). In [184] the same system is evaluated with regards to the amount of required detector. It is stated that the near-maximal coverage is achieved with between 1000 to 2000 individual detectors using the r-contiguous matching rule with r = 8, 10, and 12. In [185] the authors present the use of 13, 90, 300, 736, 1683, and 3691 detectors for r = 7, 8, 9, 10, 11, and 12, respectively. In addition, for their Hybrid Neuro-Immune System, the authors used 400 detectors for the Real-Valued Negative Selection algorithm. In [186], the authors stated that 1000 detectors covered 99.99% of their search space, while 400 detectors achieved a 99% coverage. In Chapter 4, the analysis of detector placement used 100 detectors. While there is no clear indication on how many detectors to select, it was decided to use more detectors than for the detector placement investigation but also to keep a resource limited design in mind. The problem with the values presented in [185] and [186] is that they are not based on the framework of [98]. Therefore parameters had to be arbitrarily selected based on the experience collected by the author during the research described in Chapter 4 and values highlighted in literature. The total available space for detectors is set to 400 (in accordance to [186]), which has to be split between immature and mature detectors. Mature detectors receive a higher percentage of the available space because only they can raise an alarm. Therefore the detector amount is split in a three to one ratio, resulting in maximum 300 mature detectors and 100 immature detectors.

Another aspect of interest is the immature and mature period of the detector, as these variables determine the length a detector spends in the training and test phase. In [98] these parameters were set to 4 and 14 days, respectively. In [184], the authors investigated the relationship between the immature phase and the false positive classification and reported a linear coherence that they could not explain. The problem

with the original values for these two time periods is, that a person's behaviour can change rather quickly and the selected mature phase might be too long. In an attempt to offer a preferably short false alarm time when the behaviour changes but still keep a similar ratio of immature to mature days, these values were arbitrarily selected to 2 days for the immature and 4 days for the mature phase. The mature array includes sensor information of the last 6 days in three immature generations.

The last aspect for the AIS is the minimal (min) and maximal (max) affinity radius. The min radius was an arbitrary selection for the Variable Distance Random Placement (VDRP) algorithm in Chapter 4. The max affinity radius is based upon the idea to start with a possible large affinity area, as the first step of the implemented algorithm is to reduce the affinity area to the closest immature detector in set $R_{Immature}$. A parameter that is not discussed in this section is the threshold value of activated detectors for the two AIS-based algorithms as the investigation of this parameter will be part of the comparison process described later in this chapter. The selection of the threshold value will have a direct impact on the three performance parameters precision, recall, and f-measure.

6.5 Description of the Designed Scenarios

Three different scenarios were designed to analyse the ability of the AIS-based monitoring systems to detect abnormal activity. The aim of the first two experiments is to compare AIS with other classification algorithms, while the goal of the last experiment is to investigate the long-term stability of the AIS algorithm itself. Each scenario was generated using the simulation environment in combination with sensor data collected by the OHDAD presented in Chapter 3. The different algorithms were set up with the parameters described in Section 6.4 (see Table 6–2 and Table 6–3).

6.5.1 Scenario 1: General Ability to Detect Abnormal Activity

The first scenario investigates the general ability for AIS to be used for the detection of abnormal activity. The other classification algorithms are thereby used as a reference. The collected sensor data consists of 5 individual sets for each activity, which need to be divided between a training and test set for the supervised classifiers (KNN, Naïve Bayes, J48, and SMO). There exists a trade-off with regards to the activity sample distributions between these two sets that needs to be considered. More samples in the training set allow for a better generalisation of the sensor data and most likely better classification

results, while more samples in the test set allow for a diverse test of the classification algorithm. The author decided to split the available data four to one, hence presenting a more general data set to the supervised classification algorithm with the aim of achieving better classification performance.

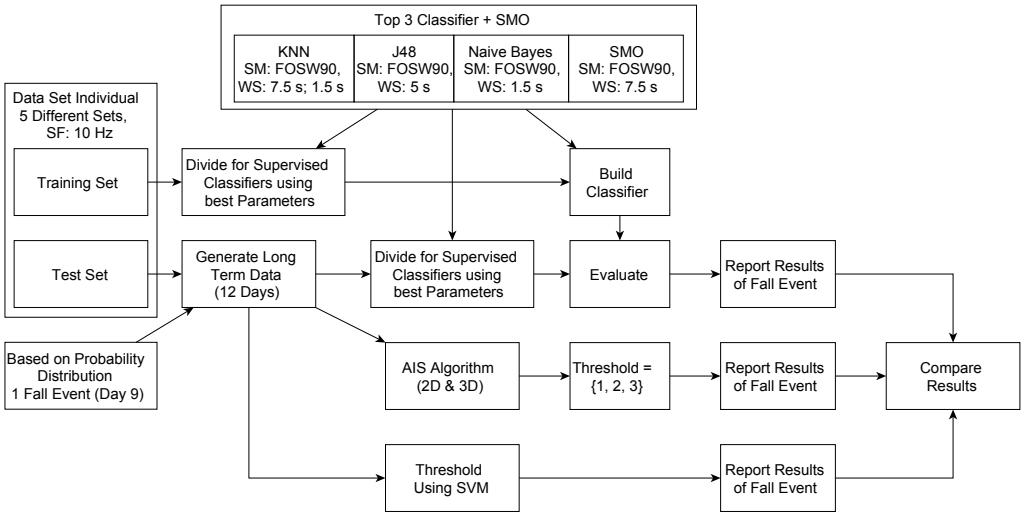


Figure 6–2: Experimental design for the classification performance evaluation of AIS

Figure 6–2 illustrates the experiment carried out for scenario 1. The simulation environment was used to generate long-term data based on the state transition presented in Table 6–4 probabilities, which were chosen to result in one fall event in the test data. The generated data is presented to the different algorithms for the classification. The threshold and the two AIS-based monitoring systems will not require any further pre-processing and use a FNSW segmentation method with a 10 s window size, to process the data. For the later comparison, the three performance measures precision, recall, and f-measure, as well as the computational load of the different algorithms are recorded.

Besides the provision of a training set, the supervised classifiers also require the extraction of the following parameters before hand: Root Mean Square (RMS), Mean, Signal Magnitude Area (SMA), Signal Vector Magnitude (here SMV), Energy, Entropy, FFTPeak, and Standard Deviation (STD). The usage and calculation of these features can be found in Chapter 5. The training and long-term test data are processed based on the selected parameter combinations, presented in Table 6–2 in Section 6.4. The training feature set is used to build the five different classifier models (KNN with 7.5 s, KNN with 1.5 s, J48 with 5 s, Naïve Bayes with 1.5 s, and SMO with 7.5 s) and evaluated using the test feature set extracted from the long-term data. The recordings of TP, FP, and FN will

be used to calculate precision, recall, and f-measure and used for the performance comparison in addition to the CL of the different classifier algorithms.

Table 6–4: State probabilities for the long-term data simulation

Transition		Transition	Simulation Day		
			Day 1 - 8	Day 9	Day 10 - 12
			Timeslot	Timeslot	Timeslot
From	To	Number	0 - 24	0 - 24	0 - 24
Stand		1	0.1	0.1	0.1
Stand	Sitting Down	2	0.3	0.3	0.3
Sitting Down	Sitting	3	1	1	1
Sitting		4	0.7	0.7	0.7
Sitting	Getting Up	5	0.3	0.3	0.3
Getting Up	Stand	6	1	1	1
Stand	Lying Down	7	0.2	0.2	0.2
Lying Down	Lying	8	1	1	1
Lying		9	0.85	0.85	0.85
Lying	Getting Up from Lying	10	0.15	0.15	0.15
Getting Up from Lying	Stand	11	1	1	1
Stairs Up	Walking	12	0.2	0.2	0.2
Stairs Up		13	0.8	0.8	0.8
Walking	Stairs Up	14	0.1	0.1	0.1
Walking	Stairs Down	15	0.1	0.1	0.1
Stairs Down		16	0.8	0.8	0.8
Stairs Down	Walking	17	0.2	0.2	0.2
Walking	Stand	18	0.3	0.3	0.3
Stand	Walking	19	0.4	0.4	0.4
Walking		20	0.5	0.499	0.5
Walking	Fall	21	0	0.001	0
Fall	Lying on Ground	22	1	1	1
Lying on Ground		23	0.65	0.65	0.65
Lying on Ground	Get Up from Ground	24	0.35	0.35	0.35
Get Up from Ground	Stand	25	1	1	1

The activity fragments used for this scenario have one limitation. Section 6.4 highlighted the existence of activities that could be incorrectly classified by the threshold-based algorithm as a fall because of their high SVM value. Therefore, these activity fragments were deliberately used for the training set of the supervised classifiers. The aspect of having similar looking activates is considered independently in scenario 2.

6.5.2 Scenario 2: Detection of Abnormal Activity with Similar Looking Activities

Scenario 2 is designed to investigate the detection of abnormal activity under the condition that activities look similar. The aim of the experiment is to investigate whether the AIS-based algorithm can disregard a frequent normal event, which has a similar characteristic to an abnormal activity. Hence, this scenario is an adaptation of the first scenario with three small changes. The first is the exclusion of the threshold algorithm as several normal activities exceed the threshold value and will cause an increased false alarm rate. The second change is the exclusion of the analysis of computational load, as

this factor is independent of the training and test set used. The third modification is that the corresponding fragments with the highest SVM value replace the sitting down and lying down activity for the long-term data generation.

6.5.3 Scenario 3: AIS Long-Term Stability

The goal of scenario 3 is to evaluate the long-term stability of the AIS-based monitoring system. The aim is to investigate, if an extensive runtime results in problems with the detection of abnormal activity, when the algorithm is run for an extensive amount of time. This analysis is required, as the literature involved with the framework used did not cover the issue of convergence and the outcome of $AIS_{Runtime} \rightarrow \infty$ is therefore unknown. For scenario 3 the adaptation to the first scenario are an increased runtime of 90 days with one abnormal activity during day 75. Furthermore, this scenario excludes the analysis of the threshold algorithm and the supervised classifiers as their TP, FP, and FN rates depend on the learned model that is independent of time.

6.6 Evaluation of Results

6.6.1 Scenario 1: General Ability to Detect Abnormal Activity

This section reports on the performance of the SVM-based threshold algorithm, supervised classifiers with optimal parameter selection, and the two different AIS-based monitoring algorithms with regards to their ability to detect abnormal activity. The three performance measures from all methods tested are presented in Table 6–5. Each approach is individually assessed, followed by a graphical comparison of the best performing algorithms. At the end of this section the computational load will be compared.

The precision value of the SVM-based threshold algorithm indicates the generation of no FP (no false alarms). The 50% for the recall parameter indicates that only half of the abnormal activity was identified. Further investigation revealed that the FNSW segmentation method divided the fall event in two windows. The high SVM value is only present in the second data window, resulting in the first window not exceeding the threshold value. This highlights a general problem with the data labelling of abnormal events, as it raises the question of how to define an abnormality or a normal activity. This is similar to [89], who stated that even the classification of the same activity showed variation when labelled by a test subject and confirmed by [147], stating that

abnormalities are ill-defined. For this experiment, any recall value above 0% highlights the correct fall detection, as the generated data set only includes one fall.

Table 6–5: Performance result for test scenario 1

Method		Activity	Precision	Recall	F-Measure
Threshold	SVM	Abnormal	100.00%	50.00%	66.67%
Traditional Classifier Activities Labelled 1 - 13	KNN 7.5s	Fall	4.26%	55.56%	7.92%
	J48 5s	Fall	0.05%	45.95%	0.10%
	NB 1.5s	Fall	0.08%	7.69%	0.15%
	KNN 1.5s	Fall	0.24%	43.41%	0.48%
	SMO 7.5s	Fall	100.00%	40.74%	57.90%
Traditional Classifier Activities Labelled Normal / Abnormal	KNN 7.5s	Abnormal	4.26%	55.56%	7.92%
	J48 5s	Abnormal	88.24%	40.54%	55.56%
	NB 1.5s	Abnormal	1.69%	44.51%	3.26%
	KNN 1.5s	Abnormal	0.24%	43.41%	0.48%
	SMO 7.5s	Abnormal	100.00%	11.11%	20.00%
Method		Threshold	Precision	Recall	F-Measure
AIS	2D with XY YZ ZX	1	33.33%	100.00%	50.00%
		2	25.00%	50.00%	33.33%
		3	33.33%	50.00%	40.00%
	3D with XYZ	1	0.04%	100.00%	0.08%
		2	40.00%	100.00%	57.14%
		3	33.33%	50.00%	40.00%

As highlighted in the scenario description in Section 6.5.1, the results for the supervised classifiers include the original thirteen activity labels and the two-class problem. For better readability, Table 6–5 contains only the fall activity (for the set of thirteen activities) and abnormality (for the two-class problem). The first observation is, that the change to a two-class problem improves the performance of the J48 and Naïve Bayes classifiers. The KNN classifier does not improve with the different labels, while the recall performance of the SMO algorithm actually decreases. The second observation is that all classifier methods correctly classify the fall for both activity labels. However, the third observation shows a varying level of success for the precision parameter; especially KNN and Naïve Bayes have a high FP rate and hence only achieve a precision below 5%. It is interesting to see that the SMO classifier, which was not part of the dominating classifiers on the Pareto curve in Chapter 5, has a FP rate of 0% for both activity labels. This is followed by J48, which achieves a precision of 88% for the two-class problem. The achieved recall performance of around 50% can be attributed to the wide label of abnormality and not just the short duration of the actual fall.

For the two AIS-based algorithms (2D and 3D representation) only the detection of non-self (abnormality) is reported. For each of the two algorithms, different detector activation thresholds are tested to reduce FP alarms. This aspect corresponds to the threshold function block that was introduced in Chapter 4. The range under test is limited to $Threshold = \{1, 2, 3\}$ as a high value will have a negative effect on the recall parameter. The AIS achieves recall values of 100% to 50%, similar to the SVM threshold algorithm. The precision parameter on the other hand, indicates a much lower performance. The FP segments were therefore further investigated.

The FP segments, classified by the AIS system with a $Threshold = \{2, 3\}$, are data segments, which directly follow a fall event. This means that the falsely classified activities are 'lying on the ground' and 'getting up from the ground'. The problem is that similar looking activities are not included in the frequent activity set and thereafter, based on the limitation that normality is required to be a frequent event, considered as an abnormality. Two possible solutions would be modifying the 'lying in bed' activity to include the test subject to rest on all sides in bed or including the 'lying on the ground' activity as an abnormality under the assumption that body monitoring during sleeping periods does not take place.

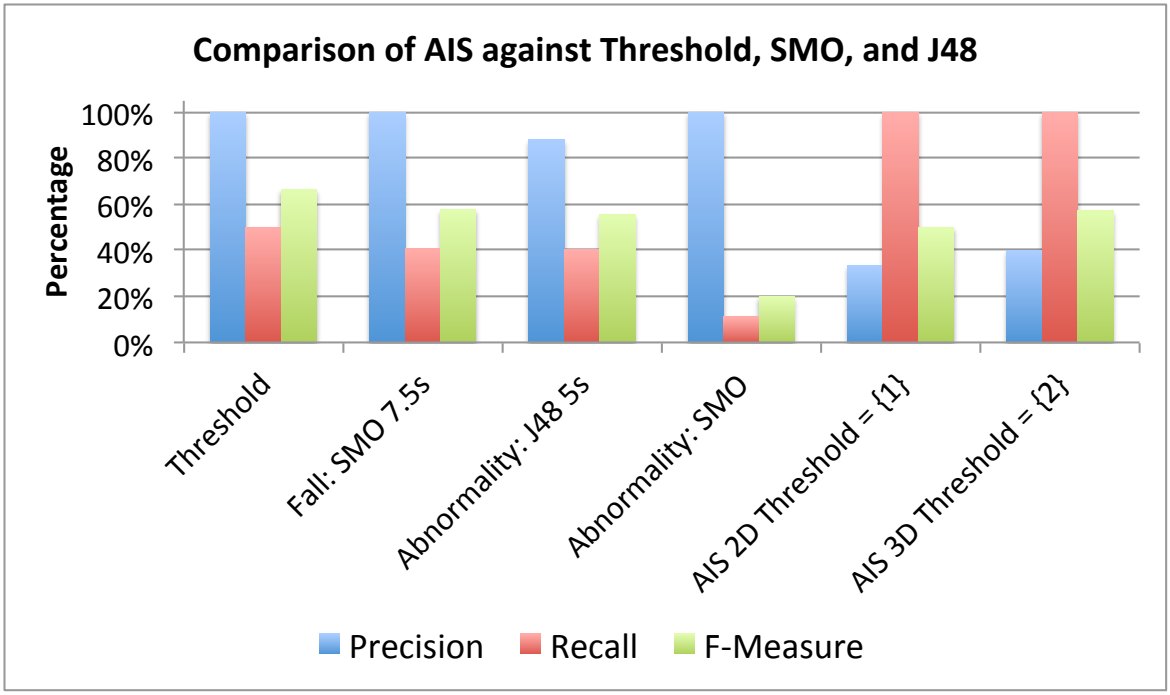


Figure 6–3: Graphical presentation of the classification performance results for test scenario 1

Figure 6–3 illustrates only the best performing classification methods from Table 6–5 for a graphical comparison. The evaluation includes the SVM-based threshold algorithm, the

SMO classifier identifying fall and abnormality, the J48 classifier for abnormality, and the two AIS-based algorithms. The AIS-based system includes the 2D representation with $Threshold = \{1\}$ and the 3D representation with $Threshold = \{2\}$. The first observation is that AIS has the worst precision performance (33.3%) of the best performing algorithms. Though FP segments follow directly a correctly classified fall event. Furthermore, AIS is the only system that does not require a-priori knowledge of the data set, as it learns over time. Though it requires an understanding of how a possible abnormal activity will affect the sensor data and how it can be detected. A possible information source would be professional care personnel, who can provide their expert knowledge during the initial design stage of the AIS-based monitoring system. The recall parameter shows mostly values of around 50% for all methods, besides SMO for the two-class problem, which misses significantly more abnormal data segments. Comparing the results based on the f-measure show that the AIS system for the 3D representation is the third best algorithm with a $Threshold = \{2\}$. Only slightly outperformed by SMO (trained with all thirteen activities) and the SVM-based threshold algorithm. In addition, all AIS-based algorithms show an advantage over the SMO classifier that is trained to distinguish between normal and abnormal data.

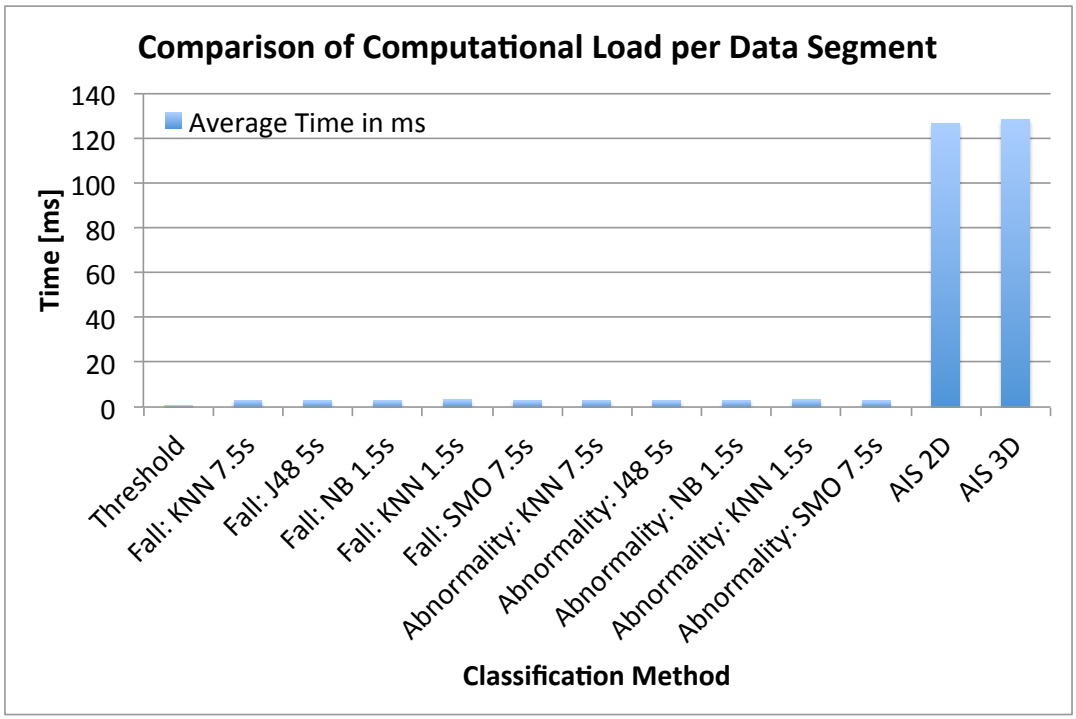


Figure 6–4: Comparison of computational load per data segment

Another aspect used to compare the algorithms, is the Computational Load (CL) that is required to carry out the data classification. Therefore, the average computational time

required to process one data segment is calculated. For the supervised classifiers this time includes the data segmentation, feature extraction and the data classification but excludes the time required to build the classifier model (SMO, KNN, J48, and Naïve Bayes) as it is only required once. Figure 6–4 shows the results of the computational load. The first observation is that the fastest algorithm is the SVM-based threshold algorithm, followed by SMO, Naïve Bayes, J48, and KNN. The AIS algorithms require a significantly longer execution time. Two factors have a significant impact on CL. A small increase in CL can be attributed to the fact that the AIS system is written in Matlab and not optimised for execution time, while SMO, Naïve Bayes, J48, and KNN are executed as Java code from Weka. The second factor is most likely the main reason for higher CL. The AIS algorithm requires the constant comparison of AIS detectors with each other and the self-set S and S^* to determine detector overlap and data matching, respectively. The execution time is therefore directly linked to the amount of used detectors, sampling frequency and window size for the sensor data.

6.6.2 Scenario 2: Detection of Abnormal Activity with Similar Looking Activities

This section analyses the effect on the detection of abnormal activity, when the abnormality is not presented by the highest SVM value. For the used SVM-based threshold algorithm, this scenario results in a high false alarm rate, as it cannot distinguish between a genuine fall and similar looking (in the sense of data representation) activities. This sensitivity is a crucial aspect of the system's acceptability as Sixsmith *et al.* reports that false alarms can quickly become intrusive and consequently results in the loss of confidence in a monitoring system by the user [6].

Table 6–6: Performance result for test scenario 2

Method		Threshold	Activity	Precision	Recall	F-Measure
AIS	2D with XY YZ ZX	1	Abnormal	20.00%	50.00%	28.71%
		2	Abnormal	20.00%	50.00%	28.71%
		3	Abnormal	20.00%	50.00%	28.71%
	3D with XY YZ ZX	1	Abnormal	0.02%	100.00%	0.03%
		2	Abnormal	0.02%	50.00%	0.04%
		3	Abnormal	20.00%	50.00%	28.57%
SMO 7.5s			Fall	100.00%	55.56%	71.43%
			Abnormal	88.24%	55.56%	68.18%

Table 6–6 illustrates the performance results for the two AIS-based algorithms and the SMO classifier method. The other classifiers are not presented due to their low f-measure.

The significant observation is that the use of similar looking activity fragments does not affect the detection of the abnormal activity for both AIS-based monitoring algorithms (recall parameter > 0). The 2D version achieves 20% precision for all thresholds. For the 3D version, the results are significantly different. The low precision measure indicates that more often mature detectors are activated by the new activities. The higher acceleration activities (sitting and lying down) can be considered as noise, which can be compensated by increasing the mature detector activation threshold. Excluding the two noisy positions, the results of the f-measure parameter do not vary from the 2D representation. The SMO algorithm still achieves 100% precision for the classification of the fall but only 88% for the abnormality detection, with a recall performance of around 50% and an f-measure of around 70%. This however, is still outperforming the AIS algorithm on the basis that the activities labelled as normal and following the fall are classified as an abnormality.

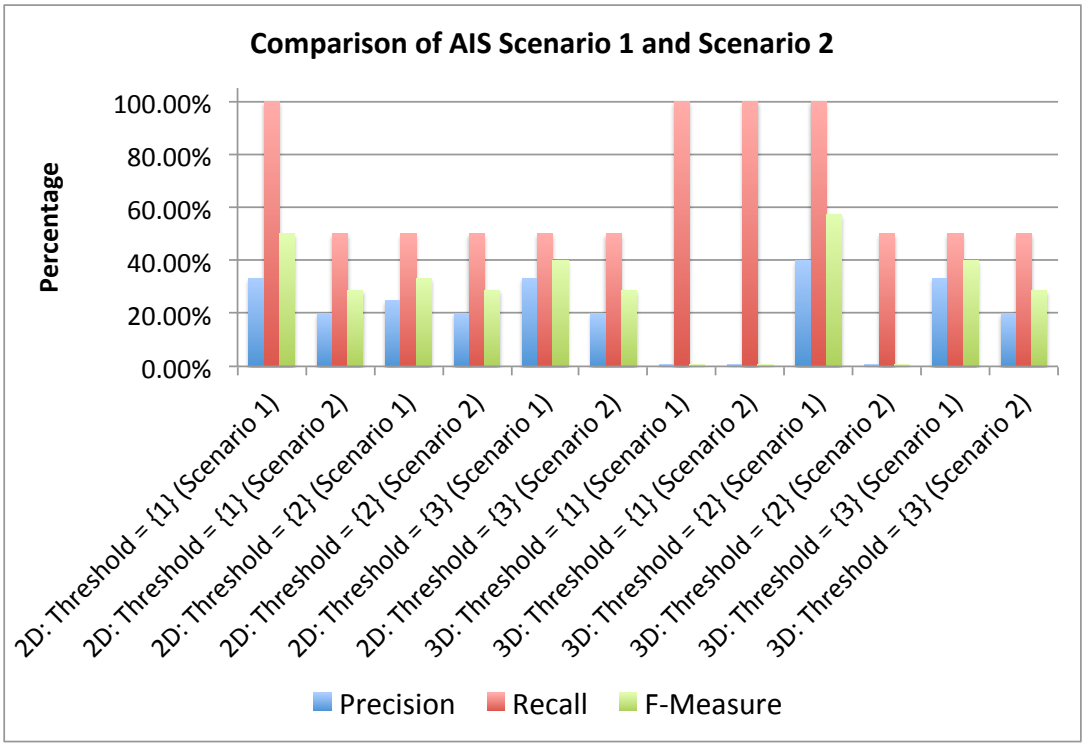


Figure 6–5: Comparison of classification performance results of AIS between scenario 1 and 2

Figure 6–5 illustrates the results achieved by the AIS-based monitoring algorithms in scenario 1 and scenario 2 for a performance comparison. For the 2D representation a

decrease in precision can be observed for scenario 2 compared to scenario 1. The recall parameter indicates that in both scenarios the AIS only achieves a 50% recall performance, as the first segment of the abnormality is difficult to detect due to the activation of a single detector. Therefore, the f-measure highlights the same performance decrease that is already apparent in the precision parameter. For the 3D representation, the decrease in precision is most significant for a $Threshold = \{2\}$. For $Threshold = \{3\}$ the decrease is less severe and comparable to the decrease witnessed for the 2D representation. This experiment showed that the use of similar looking activities results in the activation of detectors outside the defined abnormal data segments. Especially the 3D representation shows an increased FP rate compared to the 2D representation. Here the distributed monitoring of the 2D representation has an advantage in robustness, as an abnormality has to be detected in each of the three monitoring blocks simultaneously. The experiment also showed that the threshold parameter could be used to reduce the FP rate.

6.6.3 Scenario 3: AIS Long-Term Stability

The third scenario investigates the stability of the AIS-based algorithms over a long period of time. The long runtime is not a problem for the tested supervised classifiers, as their data models are not modified after the initial build using the provided training set. However, the AIS system is constantly evolving and changing. Every time a detector is removed from the detector set $R_{Immature}$ or R either age related or replaced, the search space coverage changes. Even the creation of a new random detector changes the search space coverage and increases uncertainty, as it can replace an older detector in an effort to improve the search space.

The question that this scenario therefore tries to answer is: how does the system perform in the long-term? Does the constant detector generation result in a steady state that prevents the detection of abnormal activity? With a system runtime of 90 days and abnormality occurring in day 75, the AIS system will have performed several iterations of detector life cycles before the occurrence of the fall. The system will have generated 37 immature detector generations and replaced 18 mature detector generations, based on the parameters for the AIS immature and mature time selected in Table 6–3 in Section 6.4.

Table 6–7: Performance result for test scenario 3

Method		Threshold	Precision	Recall	F-Measure
AIS	2D with XY YZ ZX	1	1.55%	100.00%	3.05%
		2	3.33%	66.67%	6.35%
		3	10.00%	66.67%	17.39%
	3D with XY YZ ZX	1	0.01%	100.00%	0.02%
		2	0.07%	33.33%	0.14%
		3	33.33%	33.33%	33.33%

The performance results presented in Table 6–7 show the correct operation of the AIS system after several detector iterations. The 2D representation achieves a classification precision of under 4% for a threshold of 1 and 2 and a precision of 10% for a threshold of 3. The 10% precision for the AIS method using a threshold of 3 can be attributed to the activity fragments following the fall event, as seen in scenario 1 and 2 earlier. The probability used for the simulation environment resulted in a prolonged time of the subject lying on the ground. The outcome is a significant lower precision rate, as the following data segments generate more FP segments. The recall parameter highlights the correct classification of the abnormality; only miss-classifying the first data segment. This results in an f-measure of below 6.3% for a threshold of 1 and 2 and around 17% for a threshold of 3. However, the outcome is in line with the earlier experimental designs.

The AIS system using the 3D data representation achieves a low precision for $Threshold = \{1,2\}$ with less than 1% and with a $Threshold = \{3\}$ the precision increases to 33%. The outcome for f-measure is only relevant for a $Threshold = \{3\}$ with 33%, while the other two threshold values achieve less than 1%.

Figure 6–6 illustrates the graphical comparison of the three performance parameters for scenario 1 and scenario 3. For the 2D AIS representation, the first observation is a significant decrease of precision and f-measure between scenario 1 and 3. For the recall measure, a light increase can be noted, which is the result of detecting 2 out of 3 abnormal data segments, while the earlier simulation only detects 1 out of 2 segments. Similar observations can be reported for the 3D representation. The performance measures show a significant decrease for $Threshold = \{1,2\}$, while the precision value for $Threshold = \{3\}$ is similar to the result in scenario 1.

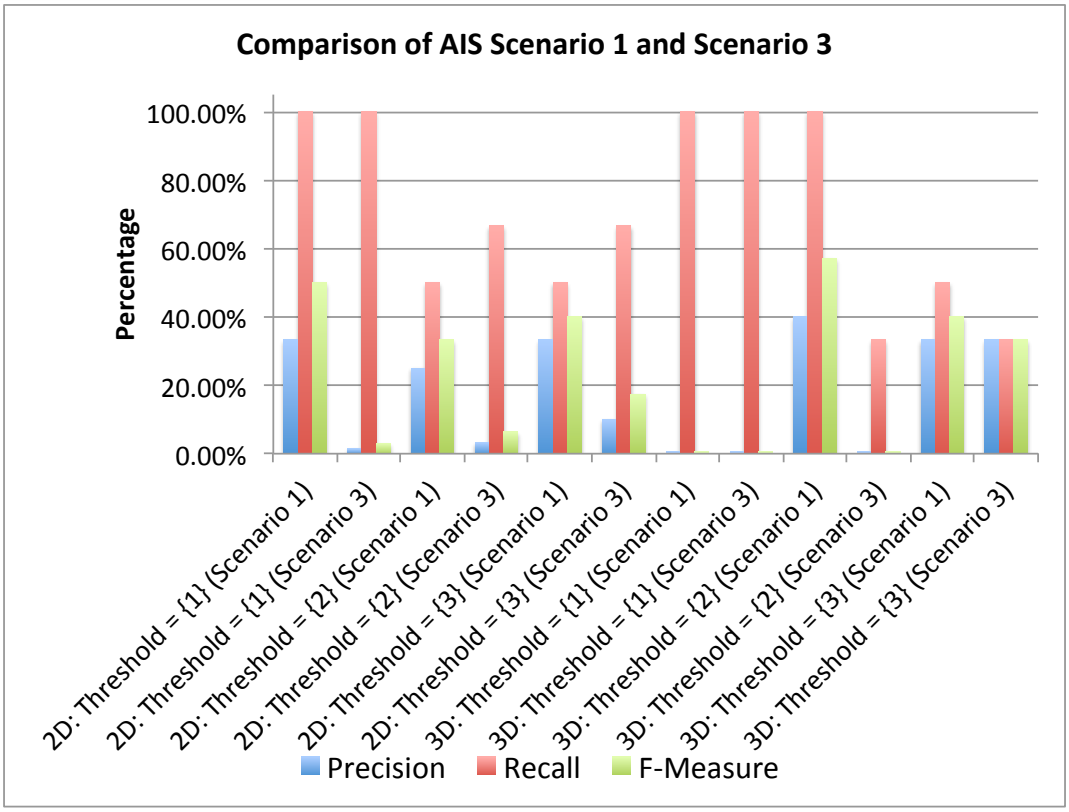


Figure 6–6: Comparison of classification performance results of AIS between scenario 1 and 3

In conclusion, the results of this scenario present that the AIS-based monitoring algorithms are stable and do not reach a steady state. Even after a prolonged runtime, both data representations (2D and 3D) detect the abnormal activity. Though, the poor performance for precision and therefore a general poor f-measure can be attributed to a prolonged "lying on the floor" activity generated by the simulation environment. These FP segments are in line with the observed results from scenario 1 and 2 and do not indicate a worth performance for the AIS when used for longer runtimes.

6.7 Discussion

This section discusses the experimental results recorded in scenario 1, 2, and 3. The experiments were designed to test different aspects of the developed AIS-based monitoring algorithms. A set of Signal Magnitude Vector (SVM)-based threshold algorithm and four supervised classifiers (NB, KNN, J48, SMO) provided a performance baseline to decide on recommendation for or against AIS as a useful algorithm in the area of Ambient Assisted Living (AAL).

From the aspect of algorithm complexity the SVM-based threshold is the best approach. It also achieves the best results in the condition that the abnormality (fall) has the highest acceleration peak, as tested in scenario 1. Two additional advantages are that a

prolonged runtime wont change the algorithm's ability to function correctly and its low computational load. The disadvantage of the algorithm is the required a-priori knowledge of the threshold value. Furthermore, it is not always given that the abnormality is presented by the highest activity peak, as tested in scenario 2. This algorithm also highlighted problems in the labelling of Activities of Daily Living (ADL). If an activity is split during the monitoring process, not all segments might be correctly classified as abnormality. This can be attributed to the fact that the required acceleration peak is relatively short compared to the remaining activity. Improvements to this kind of monitoring algorithm would be to include an additional threshold to reduce possible false alarms.

For the used supervised classification algorithms the experimental setups showed mixed results. From the four tested algorithms, KNN and Naïve Bayes achieved constantly low results with a precision and f-measure below 1%, while J48 and SMO achieved very good results between 100% - 80% during scenario 1. In scenario 2 only the SMO classifier managed to achieve good results, with the other three methods staying below 5% for precision and f-measure. The cause of these low performance measures is hard to determine as the other activities showed good classification results. A further disadvantage is the requirement of a-priori activity data to generate a training set to be able to build the classifier model. Thereafter the models are ready to be used with the certainty that even after a prolonged operational time classification will be consistent, as the models do not change over time.

The main aspect of the experiments carried out in this chapter is to determine the usefulness of the AIS-based monitoring algorithms for the detection of abnormal activity in AAL environments. Over the three different scenarios the AIS-based algorithms provided mixed results. On one hand each fall was correctly classified during the three tested scenarios. On the other hand the activities that are directly following a fall event are infrequent normal activities and are classified as abnormal. It is open for discussion, if the 'lying on the ground' activity should be considered a normal or abnormal activity. The fact that a subject is lying on the ground is abnormal but the used sensor data (accelerometer) does not allow the differentiation between a subject lying on the bed, couch, or ground. Similar misclassifications cannot be ruled out, as one of the requirements of the used AIS framework states that normal behaviour has to be represented by frequent events and abnormal behaviour by infrequent events. While the

results are not perfect in the sense of false positive data segments, the results are supporting the use of AIS-based algorithms in the area of AAL. The advantage of good classification results without a-priori knowledge is traded for a significant increased in CL. The constant comparison of elements of detector set $R_{Immature}$, R and data set S^* is computational intensive though still manageable in real-time.

The experimental design considered also the impact of the detector activation threshold. For all three scenarios, the two AIS-based algorithms showed a high FP rate when $Threshold = \{1\}$ and a reduced recall value when the $Threshold$ value is increased, highlighting a trade-off between these parameters. Under normal circumstances not missing an abnormality is more important than a false alarm, hence a low threshold should be chosen. Something that was not stated in the analysis of the three scenarios was the AIS detector activation level.

Table 6–8: Number of activated AIS detectors by the abnormal activity during scenario 1, 2, and 3

	AIS			
	2D			3D
	XY	YZ	ZX	XYZ
Scenario 1	11	15	13	12
Scenario 2	9	16	16	17
Scenario 3	7	10	19	12
Min	7	10	13	12

The analysis of activation levels showed that the first data segment of a fall event only activated one mature detector. This explains the drop in recall performance when a higher threshold of 1 is selected. Table 6–8 shows the activation levels for the second data segment. From the presented table, three observations can be made. Firstly, the activation level in the 2D representation varies based on the axes combination. Secondly, the activation level for the 2D and 3D representation are different. Thirdly, the minimal activation level is significant higher than the threshold of 3, as tested in the different scenarios. Accepting that missing the first data segment is necessary to achieve maximum precision and using the a-priori knowledge of the minimum activation levels per acceleration axis, the three performance measures are recalculated for scenario 1 and 2 and compared to the earlier results in Table 6–9.

Table 6–9: Performance results of scenario 1 and 2 with a-priori knowledge for AIS threshold

	Method		Activity	Precision		Recall		F-Measure	
				Old	New	Old	New	Old	New
Scenario 1	Threshold		Abnormal	100.00%		50.00%		66.67%	
	SMO 7.5s		Fall	100.00%		40.74%		57.90%	
	J48 5s		Abnormal	88.24%		40.54%		55.56%	
	SMO 7.5s		Abnormal	100.00%		11.11%		20.00%	
	AIS	2D with XY YZ ZX	Abnormal	33.33%	100.00%	100.00%	50.00%	50.00%	66.00%
		3D with XYZ	Abnormal	40.00%	50.00%	100.00%	50.00%	57.14%	50.00%
Scenario 2	SMO 7.5s		Fall	100.00%		55.56%		71.43%	
			Abnormal	88.24%		55.56%		68.18%	
	AIS	2D with XY YZ ZX	Abnormal	20.00%	100.00%	50.00%	50.00%	28.71%	66.00%
		3D with XYZ	Abnormal	20.00%	50.00%	50.00%	50.00%	28.57%	50.00%

The new performance measures show a significant improvement over the old parameters. For scenario 1 the new *Threshold* parameters help to outperform SMO and are on par with the SVM-based threshold algorithm. For scenario 2, the results are slightly mixed. The 2D AIS representation is on par or even outperforming the SMO classifier for the precision though beaten for recall and f-measure. This however highlights the potential AIS-based algorithms have in the area of AAL.

6.8 Summary

The performance measures used for all three scenarios were precision, recall, f-measure. In addition, the required Computational Load (CL) for the different classification methods was compared during the analysis of scenario 1. The comparison of CL showed that the SVM-based algorithm is the fastest, followed by the supervised classifiers and with significant increase the AIS-based monitoring algorithm. The further analysis of scenario 1 showed good results for the SVM-based algorithm, SMO, J48, and the two AIS-based algorithms (2D and 3D representation). KNN and Naïve Bayes did not produce satisfying results for the whole experiment. During scenario 2 only SMO and the AIS-based algorithms showed good results. In scenario 3 the correct operation of the AIS-based algorithm over a prolonged runtime was highlighted.

The three different scenarios showed that SMO and the AIS-based algorithms are useful approaches for the detection of abnormal activity in AAL environments. Though the initial results of AIS were below the ones of the SMO classifier and the author's expectation. In an effort to improve the performance of the AIS-based algorithm, the detector activation threshold was increased in order to reduce the false alarm rate. The new threshold requires more detectors to be activated with the effect that the precision of the algorithm increases to a satisfying level and the first data segment of the fall is

missed. However, this is not a significant problem, as the second segment of the fall is still detected. Tuning this parameter brings the performance of the AIS-based algorithm on par and even sometimes outperforms the SMO results. The performance of AIS comes with the down side of an increased CL compared to SMO. While it takes significantly longer to process one data segment, it is still handled fast enough to allow the real-time data analysis with the AIS-based algorithm. Furthermore, the main advantage that the AIS-based algorithm has over the SMO approach is that it does not require a data training set from the monitored person in order to generate a data model. The data model of the AIS-based algorithm is dynamically generated and functions well over a long period of time, as tested in scenario 3. The results presented in this chapter are supportive and encouraging of the author's decision to introduce the use of AIS for the detection of abnormal activity in the Ambient Assisted Living (AAL) environments, especially when a-priori knowledge is not available.

7 Discussion, Contribution, Future Work, Conclusion

7.1 Introduction

The work presented in this thesis was motivated by the needs in terms of caring for the elderly due to the increased life expectancy and the resulting demographic changes. The goal of the presented research was to introduce a novel method to the detection of abnormal activity in the context of long-term monitoring of elderly. The author therefore investigated the use of Artificial Immune Systems (AIS) using on-body accelerometer sensor data. Through the comparison with other classification algorithms, the validity of using AIS in the field of Ambient Assisted Living (AAL) was proven though areas of improvements and future research were also highlighted.

7.2 Discussion

The author believes that the work presented in this thesis has contributed to the fields of data generation, processing, analysis, classification, Artificial Immune Systems, and Ambient Assisted Living. Each area will be individually discussed in this section.

The first field discussed is the data generation, with two contributions. The first one is the Open Hardware Data Acquisition Device (OHDAD). The development of a new Data Acquisition Device was necessary after the literature review did not highlight useful data sets for the work presented here and the information of identified data acquisition devices were too limited for the reproduction. In addition, the commercial products (Real-Time Localisation System (RTLS) Ubisense and TI eZ430-Chronos discussed in Chapter 3) available to the author showed to be limited in functionality and not sufficiently accurate. In order to be useful to others, the OHDAD developed by the author had to be affordable and easy to reproduce. It was designed around mass-manufactured MEMS-sensor breakout boards and a powerful yet inexpensive microcontroller development platform. All necessary files (hardware and software) are available online and can be improved or modified to cater for different use-cases. For example, the achievable 10 Hz sampling rate of the OHDAD will be too slow for sport related activities but is adequate for the Activities of Daily Living used, as the data analysis in Chapter 5 showed only a marginal improvement in classification accuracy for higher sampling rates.

The second contribution is a simulation environment that reduces the otherwise unavoidable design lead-time and allows for a faster generation of long-term sensor data. The environment is designed around a Markov chain and transitions probabilities to simulate human behaviour over a long period of time. The generated data is used to discover a person's normal activity behaviour and its deviation from it during the data analysis stage. Another advantage this simulation approach has over the real data collection is the guarantee that the data set will contain abnormalities, which cannot be assured to happen in a real world situation. While the real human behaviour is too complex to be mathematically modelled and validated, the evaluation of this tool showed a valid approach to generate simple human activity behaviour based on short activity samples and Markov chains.

The work presented in Chapter 5 contributed to a better understanding in the data processing, analysis, and classification, which was necessary after the literature review showed an inconsistency in parameter selection when trying to identify already established classification algorithms and their corresponding parameter combinations for data segmentation, data window size, and sampling frequency. During the work presented, over 10000 different parameter combinations were tested on 23 different test subjects from two independent data sets. For a better understanding of the influence of sampling frequency, segmentation method, window size, and classification method on the Activity of Daily Living (ADL) event Classification Accuracy (CA) and Computational Load (CL), two independent ANalysis Of VAriances (ANOVA) were carried out for CA and CL. For the third analysis the results were plotted along a CA and CL axis. The Pareto-based approach can be used as a tool to chose specific parameter combinations to achieve the best CA for the lowest CL or vice versa. Unfortunately, during the experiment the importance of extracted features could not be investigated, as it would have increased the parameter variation by the factor of 2^8. The advantage would be an improved parameter selection for computational restrained embedded systems. However, the validity of the presented ANOVA results and the Pareto-based tool is not affected by it.

The work resulted in two contributions in the field of Artificial Immune Systems. The first contribution is the novel Centre Point Seeding (CPS) detector placement method. The analysis of different AIS detector placement methods showed that a random detector placement with a fixed affinity area, can generate detectors too close to each

other and as a result of overlapping affinity areas requires the use of more detectors than necessary. Therefore, CPS identifies an area of interest so detectors can be placed around it with increasing distance and affinity area for an improved coverage of the search space. This approach has the disadvantage that the identification of several areas of interest can still result in detectors getting generated too close to each other. The second contribution is the AIS-based monitoring algorithm for the detection of abnormal behaviour in Ambient Assisted Living environments. The algorithm consists of two function blocks. The first block sits at the sensor (lowest) level. There, the AIS is used to learn the data pattern over time, similar to the Hierarchical Temporal Memory, and notifies the higher level in case of a deviation. The second function block uses this notification as an indicator to classify the data segment as abnormal or a data outlier. The analysis of the designed algorithm is focusing on human acceleration, though other sensor behaviour sources such as Passive InfraRed (PIR) room occupancy data could be used but was outside the scope of this work.

The evaluation of the AIS-based algorithms does not only contribute to the field of Artificial Immune Systems but also to the more general field of Ambient Assisted Living. To the author's best knowledge, the here presented comparison of detection accuracy and computational load with regards to the AIS-based monitoring system, SVM threshold-based algorithm and the supervised classifiers J48, KNN, Naïve Bayes, and SMO have not been presented before. The experiments carried out during the evaluation process showed that only the SMO classifier outperforms the two developed AIS-based algorithms for the detection of abnormal activity. The advantage AIS has over the supervised classifiers such as SMO, is that it constantly learns and adapts to unlabelled sensor data. The author's recommendation is therefore to use SMO when a specific abnormality, for which it is possible to provide specific training data, should be detected. If the abnormality cannot be accurately described or training data for supervised classifiers is difficult to provide, AIS-based monitoring algorithms are a good alternative, especially for complex problems that cannot mathematically be described, such as faced in AAL.

A disadvantage of the AIS-based algorithms is the required computational load, compared to the other tested classifiers. Even though all tested methods can be used for the real-time data classification, the analysis of CL showed a significant difference for the required time per data segment. The use of code optimisation and improved memory

management could help to reduce CL but was outside the scope of the here presented work.

A general problem that can be seen in other AAL related research; such as low number of test subjects, test sets, and tested ADLs is also a problem in this research work. The work with human test subjects requires ethical approval and follow up studies require significant funding, increased number of test subjects in the age of 65+, as well as the interdisciplinary work between psychology, engineering, and health care personnel.

7.3 Contribution

The work carried out has contributed the following:

1. The conception and implementation of an Open Hardware Data Acquisition Device (OHDAD), an open, easy to reproduce and inexpensive sensor data collection device that facilitates the collection of normal and abnormal human activity. The system is designed to allow the short- and long-term data collection. It connects wirelessly to smart phones and laptops for a comfortable data labelling based on the on-board time stamping.
2. The conception and implementation of a software simulation environment based on Markov chains, which allows the rapid long-term data generation using short-term data collected with the OHDAD device. The adaptable layout of Markov chains and probabilities allows the generation of flexible scenarios.
3. A novel seeding technique for detector placement in AIS has been presented. Instead of randomly generating detectors in a given search space, an area of interest is identified and used to generate detectors around this area in varying distance. To further optimise the search space coverage, the detector radius is dependent on the distance from the identified area of interest.
4. Introduction and implementation of an AIS-based algorithm for the abnormal activity detection in AAL sensor data. The algorithm is an unsupervised learning classifier that identifies the test subject's activity behaviour over time and alarms in the case of a deviation.
5. The use of AIS as an abstraction layer for the area of AAL.
6. The empirical analysis of the influence of classification method (CM), window size (WS), segmentation method (SM), and sampling frequency (SF) on the Activity of Daily Living (ADL) event classification accuracy and the computational load.

7. The identification of the rank of importance of the four tested parameters for best classification accuracy (in decreasing order of importance: CM, SM, WS and SF). The analysis also highlighted that the sampling frequency can be non-significant factor for the classification accuracy.
8. The identification of the rank of importance of the four tested parameters for lowest computational load (in decreasing order of importance: CM, WS, SF and SM).
9. A Pareto curve based tool for parameter selection when looking to achieve best classification accuracy for the lowest computational load. The Pareto curve presents dominant parameter combinations that achieved the best classification accuracy for a given computational load. The tool also allows for the selection of combinations for different hardware limitations, such as a fixed sampling frequency, certain computational load, or minimum classification accuracy.
10. Evaluation of the abnormal activity classification performance of AIS with KNN, J48, Naïve Bayes, SMO, and a simple Signal Vector Magnitude (SVM)-based threshold algorithm. The comparison showed that AIS was only outperformed by SMO.
11. Evaluation of the computational load required by the AIS, KNN, J48, Naïve Bayes, SMO and a threshold algorithm based on the SVM data feature. The analysis showed that further optimisation of the computational load is required for the AIS algorithm, as it requires significant more time for the process of a data segment.
12. The work presented in this thesis has resulted in the following publications:

Conference Proceedings

- S. D. Bersch, D. Azzi, and R. Khusainov, “Artificial Immune System - A New Approach For The Long-Term Data Monitoring in Ambient Assisted Living”, 5th International Conference on Sensor Systems and Software (S-Cube), 2014. – In Press
- S. D. Bersch, D. Azzi, R. Khusainov, and I. E. Achumba “A Open Hardware Wireless Sensor Monitoring System for Human Wellbeing Research in Ambient Assisted Living”, 5th International Conference on Sensor Systems and Software (S-Cube), 2014. – In Press

- S. D. Bersch, C. M. Chislett, D. Azzi, R. Khusainov, and J. S. Briggs, “Activity detection using frequency analysis and off-the-shelf devices: fall detection from accelerometer data,” Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on, pp. 362–365, 2011.
- S. D. Bersch, D. Azzi, and R. Khusainov, “Fall Detection using Biologially Inspired Monitoring - Artificial Immune System in the Area of Assisted Living.,” ECTA-FCTA, pp. 320–323, 2011.

Journal Publications

- S. D. Bersch, D. Azzi, R. Khusainov, I. Achumba, and J. Ries, “Sensor Data Acquisition and Processing Parameters for Human Activity Classification,” Sensors, vol. 14, no. 3, pp. 4239–4270, Mar. 2014.
- S. D. Bersch, D. Azzi, and R. Khusainov, “Artificial Immune Systems for Anomaly Detection in Ambient Assisted Living Applications,” International Journal of Ambient Computing and Intelligence, vol. 5, no. 3, pp. 1–15, 2013.

7.4 Future Work

The results presented in this thesis provide the starting point for future work. The data analysis of the generated long-term sensor data showed that AIS should be considered for future work regarding the detection of abnormal activity in AAL environments. Figure 7–1 illustrates the work presented in this thesis in the perspective of possible future work. This thesis has covered the software validation and the integration testing using a simulation environment. The next steps should be to investigate the use of AIS-based monitoring approaches in real living lab environments using volunteering participants, actors or even stunt doubles [148]. After the successful execution, this should be followed by a pilot experiment using an AAL target group (elderly) and real Ambient Intelligent Environments (AmIEs). These aspects were outside the scope of this work, because they require funding, are ethically questionable (test subject can get hurt), and require, time for the long-term data collection.

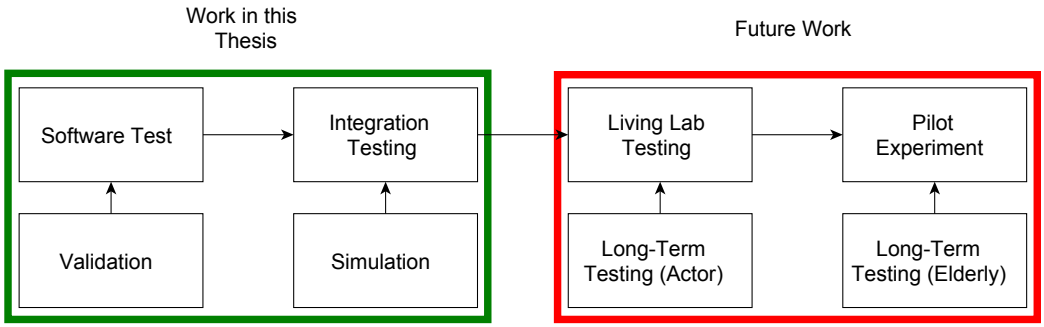


Figure 7–1: Future work in relation to carried out work (Adapted from [21])

Future work can also improve on the here presented work. The simulation environment can be improved through the inclusion of furniture and linking ADLs to the available furniture in the room. This would mean that sitting down can only occur when a chair is available in the room and the length of an activity such as walking correlates to the room dimension (e.g. walking for an hour in a room is suspicious). Another improvement would be the addition of room occupancy using an additional Markov chain. This could be used to simulate a test subject in a specific room for a given time with the addition of specific room exit and entry activities and the option to interrupt long sitting or sleeping states.

In conjunction with the above changes, future work should also investigate the performance between an AIS-based monitoring system used to detect abnormal room occupancy against the symmetrical threshold method presented in [69]. Furthermore, the integration of memory detectors in combination with user or expert co-stimulation could be used to identify and reduce false alarms. If an alarm condition is identified, user (test subject or expert) feedback could be used to improve the decision of false alarm or genuine alarm condition. Data points that are known to generate false alarms could be covered by memory detectors to prevent a full alarm condition and instead raise an intermittent alarm (traffic light conditions) in an attempt to produce less intrusive alarms, which were a problem in [6]. When using human interaction, another factor to consider as a measure of wellbeing is the time it takes between requesting feedback from the user and the actual return of the feedback. The time difference can be used as a measure for the cognitive skill or in case of an emergency correlating to the severity.

Another aspect that was not investigated by this work is the possibility of vaccination of the AIS. Like the Human Immune System, the AIS could be vaccinated through the pre-selection of memory detectors. If through extensive testing of different test subjects, a

specific location of abnormality is detected, this position could be injected like a vaccine in already deployed AIS-based monitoring systems.

One improvement to the presented AIS algorithm is the selection of the detector start radius. Instead of selecting a specific radius, Matlab provides the definition of ∞. The infinite radius would be reduced to either match the closest self-set S (in line with V-Detector presented in [159]) or the closest immature detector.

The parameter analysis presented in Chapter 5 can also be further extended. One of the critical aspects is the influence of the extracted features. Different feature combination might increase the classification accuracy for different segmentation parameter. Reducing the required feature set will automatically improve the computational load too. Investigating the four different parameters of sampling frequency, segmentation method, window size, and classifier method for each feature combination would have increased the complexity by a further 2^8 parameter levels to 2654208 possible combinations per test subject.

Reference

[1] "The 2009 Ageing Report: economic and budgetary projections for the EU-27 Member States (2008-2060)," European Commission (DG ECFIN) and the Economic Policy Committee (AWG), 2009.

[2] "The 2012 Ageing Report Economic and budgetary projections for the 27 EU Member States 2010 2060," European Commission Economic and financial affairs the Economic Policy Committee, Sep. 2012.

[3] WHO, "World Health Organization," *www.who.int*. [Online]. Available: http://www.who.int/features/factfiles/ageing/en/index.html. [Accessed: 02-Sep-2013].

[4] H. Kunemund, "Changing welfare states and the „sandwich generation"–increasing burden for the next generation," *International Journal of Ageing and Later Life*, vol. 1, no. 2, pp. 11–30, 2006.

[5] J. Nehmer, M. Becker, A. Karshmer, and R. Lamm, "Living assistance systems: an ambient intelligence approach," *Proceedings of the 28th international conference on Software engineering*, pp. 43–50, 2006.

[6] A. J. Sixsmith, "An evaluation of an intelligent home monitoring system," *Journal of telemedicine and telecare*, vol. 6, no. 2, pp. 63–72, 2000.

[7] E. Berndt and R. Wichert, "Marktpotenziale, Entwicklungschancen, gesellschaftliche, gesundheitliche und ökonomische Effekte der zukünftigen Nutzung von Ambient Assisted Living Technologien," *Ambient Assisted Living-AAL*, 2010.

[8] *The German AAL Standardization Roadmap (= Ambient Assisted Living)*. VDE Association For Electrical, Electronic & Information Technologies, 2012, pp. 1–99.

[9] E. Portmann, A. Andrushevich, R. Kistler, and A. Klapproth, "Prometheus — Fuzzy information retrieval for semantic homes and environments," *Human System Interactions (HSI), 2010 3rd Conference on*, pp. 757–762, 2010.

[10] M. Amoretti, F. Wientapper, F. Furfari, S. Lenzi, and S. Chessa, "Sensor Data Fusion for Activity Monitoring in Ambient Assisted Living Environments," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 24, no. 15, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 206–221.

[11] M. Floeck and L. Litz, "Activity- and Inactivity-Based Approaches to Analyze an Assisted Living Environment," *Emerging Security Information, Systems and Technologies, 2008. SECURWARE '08. Second International Conference on*, pp. 311–316, 2008.

[12] G. V. D. Broek, F. Cavallo, L. Odetti, and C. Wehrmann, "Ambient Assisted Living Roadmap," pp. 1–120, Sep. 2009.

[13] M. Gersch, R. Lindert, and M. Hewing, "AAL-business models: Different Prospects for the Successful Implementation of Innovative Services in the Primary and Secondary Healthcare Market," *AALIANCE conference - Malaga, Spain*, pp. 1–10, Feb. 2010.

[14] M. Segarra and F. Andre, "Building a Context-Aware Ambient Assisted Living Application Using a Self-Adaptive Distributed Model," *Autonomic*

and Autonomous Systems, 2009. ICAS '09. Fifth International Conference on, pp. 40–44, 2009.

[15] "Philips Research - Ambient Intelligence, changing lives for the better," *research.philips.com*. [Online]. Available: http://www.research.philips.com/technologies/projects/ami/background.html. [Accessed: Feb-2012].

[16] D. J. Cook, J. C. Augusto, and V. R. Jakkula, "Ambient intelligence: Technologies, applications, and opportunities," *Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 277–298, 2009.

[17] "Philips Research - The technologies of Ambient Intelligence," *research.philips.com*. [Online]. Available: http://www.research.philips.com/technologies/projects/ami/breakthroughs.html. [Accessed: Feb-2012].

[18] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[19] P. Norvig and S. Russell, "Artificial intelligence: a modern approach," *Prentice Hall*, 1995.

[20] *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US, 2010.

[21] J. A. Botia, A. Villa, and J. Palma, "Ambient Assisted Living system for in-home monitoring of healthy independent elders," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8136–8148, Jul. 2012.

[22] K. Bache and M. Lichman, "UCI Machine Learning Repository," *archive.ics.uci.edu*. [Online]. Available: http://archive.ics.uci.edu/ml/index.html. [Accessed: 25-Nov-2010].

[23] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Forster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, and A. Ferscha, "Collecting complex activity datasets in highly rich networked sensor environments," presented at the Networked Sensing Systems (INSS), 2010 Seventh International Conference on, 2010, pp. 233–240.

[24] "mHealth," *mhealth.ccs.neu.edu*. [Online]. Available: http://mhealth.ccs.neu.edu/. [Accessed: 19-Mar-2014].

[25] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," *Pervasive Computing*, pp. 1–17, 2004.

[26] B. Kaluža, V. Mirchevska, E. Dovgan, M. Lustrek, and M. Gams, "An agent-based approach to care in independent living," *Ambient Intelligence*, pp. 177–186, 2010.

[27] D. Roggen, G. Tröster, P. Lukowicz, A. Ferscha, and J. del R Millán, "Activity and Context Recognition with Opportunistic Sensor Configurations," Mar. 2010.

[28] R. Khusainov, D. Azzi, I. Achumba, and S. D. Bersch, "Real-Time Human Ambulation, Activity, and Physiological Monitoring: Taxonomy of Issues, Techniques, Applications, Challenges and Limitations," *Sensors*, vol. 13, no. 10, pp. 12852–12902, Oct. 2013.

[29] J.-I. Pan, C.-J. Yung, C.-C. Liang, and L.-F. Lai, "An Intelligent Homecare Emergency Service System for Elder Falling," in *World Congress on Medical Physics and Biomedical Engineering*, vol. 14, no. 114, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 424–428.

[30] S.-W. Lee and K. Mase, "Activity and location recognition using

wearable sensors," *Pervasive Computing, IEEE*, vol. 1, no. 3, pp. 24–32, 2002.

[31] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Pattern Recognition and Image Analysis*, Lecture Notes in Computer Science. vol. 6669, Springer Berlin Heidelberg, 2011, pp. 289–296.

[32] B. O'Flynn, P. Angove, J. Barton, A. Gonzalez, J. O'Donoghue, and J. Herbert, "Wireless biomonitor for ambient assisted living," presented at the International Conference on Signals & Electronic Systems (ICSES), 2006.

[33] "Wireless Sensor Networks Prototyping Platform_25mm | Tyndall," *tyndall.ie*. [Online]. Available: http://www.tyndall.ie/content/wireless-sensor-networks-prototyping-platform25mm. [Accessed: 19-Mar-2014].

[34] V. Gerasimov, T. Selker, and W. Bender, "Sensing and effecting environment with extremity computing devices," *Motorola Offspring*, 2002.

[35] V. Gerasimov, "Hoarder aka Swiss Army Knife (SAK) board." [Online]. Available: http://vadim.oversigma.com/Hoarder/Hoarder.htm. [Accessed: 13-Nov-2012].

[36] A. Lombardi, M. Ferri, G. Rescio, M. Grassi, and P. Malcovati, "Wearable wireless accelerometer with embedded fall-detection logic for multi-sensor ambient assisted living applications," *Sensors, 2009 IEEE*, pp. 1967–1970, 2009.

[37] T. Huynh and B. Schiele, "Analyzing features for activity recognition," presented at the Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies, 2005, pp. 159–164.

[38] M. Sekine, T. Tamura, T. Togawa, and Y. Fukui, "Classification of waist-acceleration signals in a continuous walking record," *Medical Engineering & Physics*, vol. 22, no. 4, pp. 285–291, 2000.

[39] M. Sekine, T. Tamura, T. Fujimoto, and Y. Fukui, "Classification of walking pattern using acceleration waveform in elderly people," presented at the Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE, 2000, vol. 2, pp. 1356–1359.

[40] S. J. Preece, J. Y. Goulermas, L. P. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *Biomedical Engineering, IEEE Transactions on*, vol. 56, no. 3, pp. 871–879, 2009.

[41] N. Wang, E. Ambikairajah, N. H. Lovell, and B. G. Celler, "Accelerometry Based Classification of Walking Patterns Using Time-frequency Analysis," presented at the Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE, 2007, pp. 4899–4902.

[42] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," presented at the Proceedings of the 17th conference on Innovative applications of artificial intelligence, 2005, vol. 3, pp. 1541–1546.

[43] J. Parkka, L. Cluitmans, and M. Ermes, “Personalization algorithm for real-time activity recognition using PDA, wireless motion bands, and binary decision tree,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 5, pp. 1211–1215, 2010.

[44] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, “Activity recognition and monitoring using multiple sensors on different body positions,” *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, pp. 1–4, 2006.

[45] E. K. Antonsson and R. W. Mann, “The frequency content of gait,” *Journal of biomechanics*, vol. 18, no. 1, pp. 39–47, 1985.

[46] C. V. Bouten, K. T. Koekkoek, M. Verduin, R. Kodde, and J. D. Janssen, “A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity,” *Biomedical Engineering, IEEE Transactions on*, vol. 44, no. 3, pp. 136–147, 1997.

[47] H. Gjoreski, M. Gams, and I. Chorbev, “3-axial accelerometers activity recognition,” *ICT Innovations 2010 Web Proceedings*, pp. 51–58, 2010.

[48] E. Pietka, “Expert systems in parameter extraction of the ECG signal,” presented at the Engineering in Medicine and Biology Society, 1988. Proceedings of the Annual International Conference of the IEEE, 1988, pp. 165–166.

[49] E. Keogh, S. Chu, D. Hart, and M. Pazzani, “An online algorithm for segmenting time series,” presented at the Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, 2001, pp. 289–296.

[50] C.-S. J. Chu, “Time series segmentation: A sliding window approach,” *Inform Sciences*, vol. 85, no. 1, pp. 147–173, Jul. 1995.

[51] S. Kozina, M. Lustrek, and M. Gams, “Dynamic signal segmentation for activity recognition,” presented at the Proceedings of International Joint Conference on Artificial Intelligence, 2011, pp. 15–22.

[52] J. Ortiz Laguna, A. Olaya, and D. Borrajo, “A dynamic sliding window approach for activity recognition,” presented at the UMAP'11: Proceedings of the 19th international conference on User modeling, adaption, and personalization, 2011, pp. 219–230.

[53] M. N. Nyan, F. Tay, K. Seah, and Y. Y. Sitoh, “Classification of gait patterns in the time–frequency domain,” *Journal of biomechanics*, vol. 39, no. 14, pp. 2647–2656, 2006.

[54] T. van Kasteren, A. Noulas, G. Englebienne, and Ben Kröse, “Accurate activity recognition in a home setting,” presented at the UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing, 2008.

[55] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, “Fine-grained activity recognition by aggregating abstract object usage,” *Wearable Computers, Proceedings Ninth IEEE International Symposium on*, pp. 44–51, 2005.

[56] A. Elfaham, H. Hagras, S. Helal, S. Hossain, Jae Woong Lee, and D. Cook, “A fuzzy based verification agent for the Persim human activity simulator in Ambient Intelligent Environments,” *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pp. 1–8, 2010.

[57] I. Armac and D. Retkowitz, “Simulation of Smart Environments,” presented at the Pervasive Services, IEEE International Conference on,

2007, pp. 257–266.
[58] V. Lesser, M. Atighetchi, B. Benyo, B. Horling, A. Raja, R. G. Vincent, T. Wagner, P. Xuan, and S. X. Zhang, "The Intelligent Home Testbed," *Proceedings of the Autonomy Control Software Workshop (Autonomous Agent Workshop)*, vol. 2, pp. 1–8, Jan. 1999.
[59] Y. Murakami, Y. Sugimoto, and T. Ishida, "Modeling human behavior for virtual training systems," presented at the Proceedings of the National Conference on Artificial Intelligence, 2005, vol. 20, no. 1, p. 127.
[60] S. Brailsford, J. Sykes, and P. Harper, "Incorporating Human Behavior in Healthcare Simulation Models," *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pp. 466–472, 2006.
[61] S. Brailsford and B. Schmidt, "Towards incorporating human behaviour in models of health care systems: an approach using discrete event simulation," *European Journal of Operational Research*, vol. 150, no. 1, pp. 19–31, 2003.
[62] N. Noury, "AILISA : experimental platforms to evaluate remote care and assistive technologies in gerontology," presented at the Enterprise networking and Computing in Healthcare Industry, 2005. HEALTHCOM 2005. Proceedings of 7th International Workshop on, 2005, pp. 67–72.
[63] G. Virone, B. Lefebvre, N. Noury, and J. Demongeot, "Modeling and computer simulation of physiological rhythms and behaviors at home for data fusion programs in a telecare system," presented at the Enterprise Networking and Computing in Healthcare Industry, 2003. Healthcom 2003. Proceedings. 5th International Workshop on, 2003, pp. 111–117.
[64] T. Hadidi and N. Noury, "Model and Simulator of the activity of the elderly person in a Health Smart Home," *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*, pp. 7–10, 2010.
[65] O. Lo, L. Fan, W. J. Buchanan, C. Thuemmler, A. S. Khedim, A. Lawson, O. Uthmani, and D. Bell, "Patient simulator: towards testing and validation.," 2011.
[66] D. Thomas, A. Joiner, Wei Lin, M. Lowry, and T. Pressburger, "The unique aspects of simulation verification and validation," *Aerospace Conference, 2010 IEEE*, pp. 1–7, 2010.
[67] N. Oreskes, "Evaluation (not validation) of quantitative models.," *Environmental Health Perspectives*, vol. 106, no. 6, p. 1453, 1998.
[68] G. Virone and A. Sixsmith, "Activity prediction for in-home activity monitoring," *Intelligent Environments, 2008 IET 4th International Conference on*, pp. 1–4, 2008.
[69] G. Virone, M. Alwan, S. Dalal, S. Kell, B. Turner, J. Stankovic, and R. Felder, "Behavioral Patterns of Older Adults in Assisted Living," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, no. 3, pp. 387–398, 2008.
[70] Virone, "Monitoring activity patterns and trends of older adults," *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pp. 2071–2074, 2008.
[71] O. Diegel, "Intelligent Automated Health Systems for Compliance Monitoring," *TENCON 2005 2005 IEEE Region 10*, pp. 1–6, 2005.

[72] A. Naseer, T. Eldabi, and M. Jahangirian, "Cross-sector analysis of simulation methods: a survey of defense and healthcare," *Transforming Government: People, Process and Policy*, vol. 3, no. 2, pp. 181–189, 2009.

[73] California Rangelands Rsearch and Information Center, "Types of Monitoring," *Agronomy and Range Science*, 1995.

[74] H. Storf, M. Becker, and M. Riedl, "Rule-based activity recognition framework: Challenges, technique and learning," *Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009. 3rd International Conference on*, pp. 1–7, 2009.

[75] A. U. Alahakone and S. A. Senanayake, "A real-time system with assistive feedback for postural control in rehabilitation," *Mechatronics, IEEE/ASME Transactions on*, vol. 15, no. 2, pp. 226–233, 2010.

[76] B. Moreels, "Romberg Test - Physiopedia, universal access to physiotherapy knowledge.," *physio-pedia.com*. [Online]. Available: http://www.physio-pedia.com/Romberg_Test. [Accessed: Nov-2013].

[77] R. Ali, L. Atallah, B. Lo, and G.-Z. Yang, "Detection and Analysis of Transitional Activity in Manifold Space," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16, no. 1, pp. 119–128, 2012.

[78] O. D. Lara and M. A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1192–1209, 2013.

[79] C. Zhu and Weihua Sheng, "Realtime human daily activity recognition through fusion of motion and location data," *Information and Automation (ICIA), 2010 IEEE International Conference on*, pp. 846–851, 2010.

[80] T. Teixeira, Deokwoo Jung, G. Dublon, and A. Savvides, "Recognizing activities from context and arm pose using finite state machines," *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*, pp. 1–8, 2009.

[81] P. Jantaraprim, P. Phukpattaranont, C. Limsakul, and B. Wongkittisuksa, "Improving the accuracy of a fall detection algorithm using free fall characteristics," *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*, pp. 501–504, 2010.

[82] Thuy-Trang Nguyen, Myeong-Chan Cho, and Tae-Soo Lee, "Automatic fall detection using wearable biomedical signal measurement terminal," *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 5203–5206, 2009.

[83] C. Dinh, D. Tantinger, and M. Struck, "Automatic emergency detection using commercial accelerometers and knowledge-based methods," *Computers in Cardiology, 2009*, pp. 485–488, 2009.

[84] S. Cagnoni, G. Matrella, M. Mordonini, F. Sassi, and L. Ascari, "Sensor Fusion-Oriented Fall Detection for Assistive Technologies Applications," *Intelligent Systems Design and Applications, 2009. ISDA '09. Ninth International Conference on*, pp. 673–678, 2009.

[85] C. Franco, J. Demongeot, Y. Fouquet, C. Villemazet, and N. Vuillerme, "Perspectives in home TeleHealthCare system: Daily routine nycthemeral rhythm monitoring from location data," presented at the Complex, Intelligent and Software Intensive Systems (CISIS)

International Conference on, 2010, pp. 611–617.
[86] C. Franco, J. Demongeot, C. Villemazet, and N. Vuillerme, "Behavioral telemonitoring of the elderly at home: Detection of nycthemeral rhythms drifts from location data," presented at the Advanced Information Networking and Applications Workshops (WAINA), IEEE 24th International Conference on, 2010, pp. 759–766.
[87] D. Elbert, H. Storf, M. Eisenbarth, O. Unalan, and M. Schmitt, "An approach for detecting deviations in daily routine for long-term behavior analysis," presented at the Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on, 2011, pp. 426–433.
[88] T. S. Barger, D. E. Brown, and M. Alwan, "Health-status monitoring through analysis of behavioral patterns," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 1, pp. 22–27, 2005.
[89] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *Pervasive Computing*, Springer, 2004, pp. 158–175.
[90] F. Zhou, J. Jiao, S. Chen, and D. Zhang, "A case-driven ambient intelligence system for elderly in-home assistance applications," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 2, pp. 179–189, 2011.
[91] D. Monekosso and P. Remagnino, "Behavior Analysis for Assisted Living," *Automation Science and Engineering, IEEE Transactions on*, vol. 7, no. 4, pp. 879–886, 2010.
[92] A. Arcelus, R. Goubran, H. Sveistrup, M. Bilodeau, and F. Knoefel, "Context-aware smart home monitoring through pressure measurement sequences," *Medical Measurements and Applications Proceedings (MeMeA), 2010 IEEE International Workshop on*, pp. 32–37, 2010.
[93] P. Rashidi, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe, "Discovering activities to recognize and track in a smart environment," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23, no. 4, pp. 527–539, 2011.
[94] V. R. Jakkula and D. J. Cook, "Detecting Anomalous Sensor Events in Smart Home Data for Enhancing the Living Experience.," *Artificial Intelligence and Smarter Living*, vol. 11, p. 07, 2011.
[95] V. Jyothsna, "A Review of Anomaly based Intrusion Detection Systems," *International Journal of Computer Applications*, vol. 28, no. 7, pp. 26–35.
[96] D. Simic and S. Simic, "Artificial immune system in risk of falling classification," *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, pp. 627–632, 2011.
[97] S. D. Bersch, D. Azzi, and R. Khusainov, "Fall Detection using Biologially Inspired Monitoring - Artificial Immune System in the Area of Assisted Living.," presented at the ECTA-FCTA - Proceedings of the International Conference on Evolutionary Computation Theory and Applications and the Proceedings of the International Conference on Fuzzy Computation Theory and Applications [parts of the International Joint Conference on Computational Intelligence IJCCI 2011, 2011, pp. 320–323.

[98] S. A. Hofmeyr and S. Forrest, “Architecture for an artificial immune system,” *Evolutionary Computation*, vol. 8, no. 4, pp. 443–473, 2000.
[99] J. Timmis, “Artificial immune systems---today and tomorrow,” *Natural Computing: an international journal*, vol. 6, no. 1, pp. 1–18, Mar. 2007.
[100] U. Aickelin, D. Dasgupta, and F. Gu, “Artificial Immune Systems,” in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, E. K. Burke and G. Kendall, Eds. arXiv.org, 2013.
[101] V. Golovko, M. Komar, and A. Sachenko, “Principles of neural network artificial immune system design to detect attacks on computers,” *Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), 2010 International Conference on*, pp. 237–237, 2010.
[102] L. Zhi-tang, L. Yao, and W. Li, “A novel fuzzy anomaly detection algorithm based on artificial immune system,” presented at the High-Performance Computing in Asia-Pacific Region, 2005. Proceedings. Eighth International Conference on, 2005.
[103] M. Elsadig, A. Abdullah, and B. Samir, “Immune multi agent system for intrusion prevention and self healing system implement a non-linear classification,” *Information Technology (ITSim), 2010 International Symposium in*, vol. 3, pp. 1–6, 2010.
[104] Yixin Cai, Mo-Yuen Chow, Wenbin Lu, and Lexin Li, “Evaluation of distribution fault diagnosis algorithms using ROC curves,” *Power and Energy Society General Meeting, 2010 IEEE*, pp. 1–6, 2010.
[105] D. Tarapore, A. L. Christensen, P. U. Lima, and J. Carneiro, “Abnormality detection in multiagent systems inspired by the adaptive immune system,” presented at the AAMAS '13: Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, 2013.
[106] D. Djurdjanovic, C. Hearn, and Y. Liu, “Immune systems inspired approach to anomaly detection, fault localization and diagnosis in complex dynamic systems,” presented at the GCMS '10: Proceedings of the 2010 Conference on Grand Challenges in Modeling & Simulation, 2010.
[107] M. Lehmann and W. Dilger, “Controlling the heating system of an intelligent home with an artificial immune system,” *Artificial Immune Systems, Proceedings*, vol. 4163, pp. 335–348, 2006.
[108] K. Polat, S. Günes, and S. Tosun, “Diagnosis of heart disease using artificial immune recognition system and fuzzy weighted pre-processing,” *Pattern recognition*, vol. 39, no. 2006, pp. 2186–2193, 2006.
[109] G. Gianini, M. Anisetti, A. Azzini, V. Bellandi, E. Damiani, and S. Marrara, “An artificial immune system approach to anomaly detection in multimedia ambient intelligence,” *Digital Ecosystems and Technologies, 2009. DEST '09. 3rd IEEE International Conference on*, pp. 502–506, 2009.
[110] S. D. Bersch, D. Azzi, R. Khusainov, and I. Achumba, “A Open Hardware Wireless Sensor Monitoring System for Human Wellbeing Research in Ambient Assisted Living,” presented at the 5th International

Conference on Sensor Systems and Software (S-Cube), Coventry, 2014.
[111] S. D. Bersch, D. Azzi, and R. Khusainov, "Artificial Immune System - A New Approach For The Long-Term Data Monitoring in Ambient Assisted Living," presented at the 5th International Conference on Sensor Systems and Software (S-Cube), Coventry, 2014.
[112] "Ubisense Research Package - Ubisense," *ubisense.net*. [Online]. Available: http://www.ubisense.net/en/products-and-services/products/research-packages/ubisense-research-package.html. [Accessed: 16-Mar-2014].
[113] "TI Chronos Product Page," *Texas Instruments*. [Online]. Available: http://www.ti.com/tool/ez430-chronos. [Accessed: Jan-2014].
[114] S. D. Bersch, C. M. Chislett, D. Azzi, R. Khusainov, and J. S. Briggs, "Activity detection using frequency analysis and off-the-shelf devices: fall detection from accelerometer data," *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*, pp. 362–365, 2011.
[115] *arduino.cc*. [Online]. Available: http://arduino.cc. [Accessed: 16-Mar-2014].
[116] *mbed.org*. [Online]. Available: http://mbed.org. [Accessed: 16-Mar-2014].
[117] A. K. Bourke and G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Medical Engineering & Physics*, vol. 30, no. 1, pp. 84–90, Jan. 2008.
[118] P. Kumar and P. C. Pandey, "A wearable inertial sensing device for fall detection and motion tracking," presented at the India Conference (INDICON), 2013 Annual IEEE, 2013, pp. 1–6.
[119] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan, "PerFallD: A pervasive fall detection system using mobile phones," presented at the Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on, 2010, pp. 292–297.
[120] S.-H. Fang, Y.-C. Liang, and K.-M. Chiu, "Developing a mobile phone-based fall detection system on android platform," presented at the Computing, Communications and Applications Conference (ComComAp), 2012, pp. 143–146.
[121] "ADXL345 datasheet and product info | 3-Axis, ±2 g/±4 g/±8 g/±16 g Digital Accelerometer | MEMS Inertial Sensors | Analog Devices," *analog.com*. [Online]. Available: http://www.analog.com/en/mems-sensors/mems-inertial-sensors/adxl345/products/product.html. [Accessed: 16-Mar-2014].
[122] "MEMS Gyro | Gyroscope | Motion Plus | Processing - ITG-3200," *invensense.com*. [Online]. Available: http://www.invensense.com/mems/gyro/itg3200.html. [Accessed: 16-Mar-2014].
[123] "HMC5883L_3-Axis_Digital_Compass_IC.pdf," *www51.honeywell.com*. [Online]. Available: http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC5883L_3-Axis_Digital_Compass_IC.pdf. [Accessed: 16-Mar-2014].
[124] "Bosch Sensortec - BMP085," *bosch-sensortec.com*. [Online]. Available:

http://www.bosch-sensortec.com/en/homepage/products_3/environmental_sensors_1/bmp085_1/bmp085. [Accessed: 16-Mar-2014].

[125] "DS3231 Extremely Accurate I²C-Integrated RTC/TCXO/Crystal - Overview," *maximintegrated.com*. [Online]. Available: http://www.maximintegrated.com/datasheet/index.mvp/id/4627. [Accessed: 16-Mar-2014].

[126] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, and N. H. N. S. A. R. E. I. T. O. Lovell, "Barometric Pressure and Triaxial Accelerometry-Based Falls Event Detection," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 6.

[127] F. Bianchi, S. Redmond, M. Narayanan, S. Cerutti, B. Celler, and N. Lovell, "Falls event detection using triaxial accelerometry and barometric pressure measurement," *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 6111–6114, 2009.

[128] D. International, "Demystifying 802.15.4 and ZigBee," *digi.com*. [Online]. Available: http://www.digi.com/pdf/wp_zigbee.pdf. [Accessed: 16-Mar-2014].

[129] "EAGLE PCB Design Software - Schematic & Layout Editor & Autorouter," *cadsoftusa.com*. [Online]. Available: http://www.cadsoftusa.com/eagle-pcb-design-software/product-overview/?language=en. [Accessed: 16-Mar-2014].

[130] "Rapid PCB Prototyping - LPKF Laser & Electronics AG," *lpkf.com*. [Online]. Available: http://www.lpkf.com/products/rapid-pcb-prototyping/index.htm. [Accessed: 16-Mar-2014].

[131] S. D. Bersch, D. Azzi, and R. Khusainov, "OHDAD — Bitbucket," *bitbucket.org*. [Online]. Available: https://bitbucket.org/Sebastian_B/ohdad. [Accessed: 30-Apr-2014].

[132] Y. Shafranovich, "RFC 4180 - Common Format and MIME Type for Comma-Separated Values (CSV) Files," *tools.ietf.org*. [Online]. Available: http://tools.ietf.org/html/rfc4180. [Accessed: 16-Mar-2014].

[133] MathWorks, Ed., *MATLAB - The Language of Technical Computing*. [Online]. Available: http://www.mathworks.co.uk/products/matlab/. [Accessed: 02-May-2010].

[134] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. del R Millán, and D. Roggen, "The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognition Letters*, vol. 34, no. 15, Nov. 2013.

[135] "ADIS16488 datasheet and product info | Tactical Grade Ten Degrees of Freedom Inertial Sensor | MEMS Inertial Measurement Units | Analog Devices," *analog.com*. [Online]. Available: http://www.analog.com/en/mems-sensors/mems-inertial-measurement-units/adis16488/products/product.html. [Accessed: 16-Mar-2014].

[136] F. Werner, J. Diermaier, S. Schmid, and P. Panek, "Fall detection with distributed floor-mounted accelerometers: An overview of the development and evaluation of a fall detection system within the project eHome," presented at the Pervasive Computing Technologies

for Healthcare (PervasiveHealth), 2011 5th International Conference on, 2011, pp. 354–361.

[137] G. Virone, D. Istrate, M. Vacher, N. Noury, J. Serignat, and J. Demongeot, "First steps in data fusion between a multichannel audio acquisition and an information system for home healthcare," *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, vol. 2, pp. 1364–1367 Vol.2, 2003.

[138] C. Zhu and Weihua Sheng, "Recognizing human daily activity using a single inertial sensor," *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, pp. 282–287, 2010.

[139] D. Sirl, "Markov Chains: An Introduction/Review," *MASCOS Workshop on Markov chains*. pp. 1–49, 2005.

[140] C. M. Grinstead and J. L. Snell, "Introduction to probability," 2nd ed., no. 11, American Mathematical Society, 1998.

[141] R. Nocua, N. Noury, C. Gehin, A. Dittmar, and E. McAdams, "Evaluation of the autonomic nervous system for fall detection," *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 3225–3228, 2009.

[142] N. V. Chawla, "Data Mining for Imbalanced Datasets: An Overview," in *Data Mining and Knowledge Discovery Handbook*, no. 45, Boston, MA: Springer US, 2010, pp. 875–886.

[143] J. P. Kleijnen, "Validation of simulation, with and without real data," Center for Economic Research, Tilburg University, 1998.

[144] Virone, "Integration of an Environmental Sound Module to an Existing In-Home Activity Simulator," *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pp. 3810–3813, 2007.

[145] G. Virone, N. Noury, and J. Demongeot, "A system for automatic measurement of circadian activity deviations in telemedicine," *Biomedical Engineering, IEEE Transactions on*, vol. 49, no. 12, pp. 1463–1469, 2002.

[146] D. Monekosso and P. Remagnino, "Behavior Analysis for Assisted Living," *Automation Science and Engineering, IEEE Transactions on*, vol. 7, no. 4, pp. 879–886, 2010.

[147] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos, "A Survey on Ambient Intelligence in Healthcare," *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2470–2494, 2013.

[148] L. Liu, M. Popescu, M. Skubic, M. Rantz, T. Yardibi, and P. Cuddihy, "Automatic fall detection based on Doppler radar motion signature," presented at the Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on, 2011, pp. 222–225.

[149] D. Dasgupta, "Novelty detection in time series data using ideas from immunology," presented at the Proceedings of the International Conference on Intelligent Systems, 1996, pp. 82–87.

[150] L. de Castro, "Artificial immune systems: A novel paradigm to pattern recognition," *Artificial Neural networks in pattern Recognition*, pp. 67–84, 2002.

[151] D. Dasgupta, S. Yu, and F. Nino, "Recent Advances in Artificial Immune

Systems: Models and Applications," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 1574–1587, Mar. 2011.

[152] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," presented at the Research in Security and Privacy, 1994. Proceedings., 1994 IEEE Computer Society Symposium on, 1994, pp. 202–212.

[153] U. Aickelin and S. Cayzer, "The Danger Theory and Its Application to Artificial Immune Systems," presented at the Proceedings of the 1st Internat Conference on ARtificial Immune Systems (ICARIS-2002), 2008, pp. 141–148.

[154] S. A. Hofmeyr and S. Forrest, "Immunity by design: An artificial immune system," *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 2, pp. 1289–1296, 1999.

[155] Yajing Zhang, "A Novel Immune Detection Algorithm for Anomaly Detection," *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pp. 1441–1446, 2005.

[156] D. W. Bradley and A. M. Tyrrell, "Immunotronics: Hardware fault tolerance inspired by the immune system," in *Evolvable Systems: From Biology to Hardware*, vol. 1801, Springer Berlin Heidelberg, 2000, pp. 11–20.

[157] F. Gonzalez, D. Dasgupta, and R. Kozma, "Combining negative selection and classification techniques for anomaly detection," presented at the Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on, 2002, vol. 1, pp. 705–710.

[158] S. Balachandran, "Multi-Shaped Detector Generation Using Real Valued Representation For Anomaly Detection," The University of Memphis, 2005.

[159] Z. Ji and D. Dasgupta, "V-detector: An efficient negative selection algorithm with 'probably adequate' detector coverage," *Inform Sciences*, vol. 179, no. 10, pp. 1390–1406, Apr. 2009.

[160] F. Gonzalez, D. Dasgupta, and J. Gómez, "The Effect of Binary Matching Rules in Negative Selection," in *Proceedings of the Genetic and Evolutionary Com- putation Conference (GECCO)*, vol. 2723, no. 25, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 195–206.

[161] D. Dasgupta, "Advances in artificial immune systems," *IEEE Computational Intelligence Magazine*, pp. 40–49, 2006.

[162] J. M. Morse, R. M. Morse, and S. J. Tylko, "Development of a Scale to Identify the Fall-Prone Patient," *Canadian Journal on Aging/La Revue canadienne du vieillissement*, vol. 8, no. 4, pp. 366–377, 1989.

[163] A. Laboratories, "Morse Fall Scale," 18-Jul-2006. [Online]. Available: http://cf.networkofcare.org/library/Morse%20Fall%20Scale.pdf. [Accessed: 19-Feb-2014].

[164] J. Hawkins and D. George, "Hierarchical temporal memory," Technical report, Numenta, Inc, 2006.

[165] E. W. Weisstein, "Circle-Circle Intersection -- from Wolfram MathWorld," *mathworld.wolfram.com*.

[166] E. W. Weisstein, "Sphere-Sphere Intersection -- from Wolfram MathWorld," *mathworld.wolfram.com*.

[167] S. D. Bersch, D. Azzi, and R. Khusainov, "Artificial Immune Systems for Anomaly Detection in Ambient Assisted Living Applications," *International Journal of Ambient Computing and Intelligence*, vol. 5, no. 3, pp. 1–15, 2013.

[168] S. D. Bersch, D. Azzi, R. Khusainov, I. Achumba, and J. Ries, "Sensor Data Acquisition and Processing Parameters for Human Activity Classification," *Sensors*, vol. 14, no. 3, pp. 4239–4270, Mar. 2014.

[169] "MATLAB and Simulink for Technical Computing - MathWorks United Kingdom," *mathworks.co.uk*. [Online]. Available: http://www.mathworks.co.uk/. [Accessed: 16-Mar-2014].

[170] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explorations Newsletter*, vol. 11, no. 1, Nov. 2009.

[171] "IBM SPSS Software," *www-01.ibm.com*. [Online]. Available: http://www-01.ibm.com/software/analytics/spss/. [Accessed: 28-Mar-2014].

[172] I. E. Achumba, S. Bersch, R. Khusainov, D. Azzi, and U. Kamalu, "On time series sensor data segmentation for fall and activity classification," presented at the e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on, 2012, pp. 427–430.

[173] J. Farringdon, A. J. Moore, N. Tilbury, J. Church, and P. D. Biemond, "Wearable sensor badge and sensor jacket for context awareness," presented at the Wearable Computers, 1999. Digest of Papers. The Third International Symposium on, 1999, pp. 107–113.

[174] H. Si, Y. Kawahara, H. Kurasawa, H. Morikawa, and T. Aoyama, "A context-aware collaborative filtering algorithm for real world oriented content delivery service," presented at the Proceedings of ubiPCMM, 2005.

[175] P. H. Veltink, H. Bussmann, W. de Vries, W. Martens, and R. C. Van Lummel, "Detection of static and dynamic activities using uniaxial accelerometers," *Rehabilitation Engineering, IEEE Transactions on*, vol. 4, no. 4, pp. 375–385, 1996.

[176] G. H. Jin, S. B. Lee, and T. S. Lee, "Context Awareness of Human Motion States Using Accelerometer," *Journal of Medical Systems*, vol. 32, no. 2, pp. 93–100, Nov. 2007.

[177] N. Kern, B. Schiele, and A. Schmidt, "Recognizing context for annotating a live life recording," *Personal and Ubiquitous Computing*, vol. 11, no. 4, pp. 251–263, Aug. 2006.

[178] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 10, no. 1, pp. 156–167, 2006.

[179] B. Nham, K. Siangliulue, and S. Yeung, "Predicting mode of transport from iphone accelerometer data," Stanford University, 2008.

[180] "Entropy of grayscale image - MATLAB entropy - MathWorks United Kingdom," *mathworks.co.uk*. [Online]. Available: http://www.mathworks.co.uk/help/images/ref/entropy.html.

[Accessed: 17-Mar-2014].
[181] MathWorks, Ed., *Fast Fourier Transform*. [Online]. Available: http://www.mathworks.co.uk/help/matlab/ref/fft.html. [Accessed: 02-Jun-2011].
[182] S. Chernbumroong, A. S. Atkins, and H. Yu, "Activity classification using a single wrist-worn accelerometer," presented at the Software, Knowledge Information, Industrial Management and Applications (SKIMA), 2011 5th International Conference on, 2011, pp. 1–6.
[183] IBM, Ed., *IBM SPSS software - Predictive analytics software and solutions*. [Online]. Available: http://www-01.ibm.com/software/uk/analytics/spss/. [Accessed: 02-May-2010].
[184] J. Balthrop, S. Forrest, and M. R. Glickman, "Revisiting LISYS: parameters and normal behavior," presented at the Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on, 2002, vol. 2, pp. 1045–1050.
[185] F. González and D. Dasgupta, "Anomaly detection using real-valued negative selection," *Genetic Programming and Evolvable Machines*, vol. 4, no. 4, pp. 383–403, 2003.
[186] Z. Ji and D. Dasgupta, "Real-valued negative selection algorithm with variable-sized detectors," *Genetic and Evolutionary Computation–GECCO 2004*, pp. 287–298, 2004.

Appendix A

SCHEMATIC DRAWING OF THE OHDAD HARDWARE

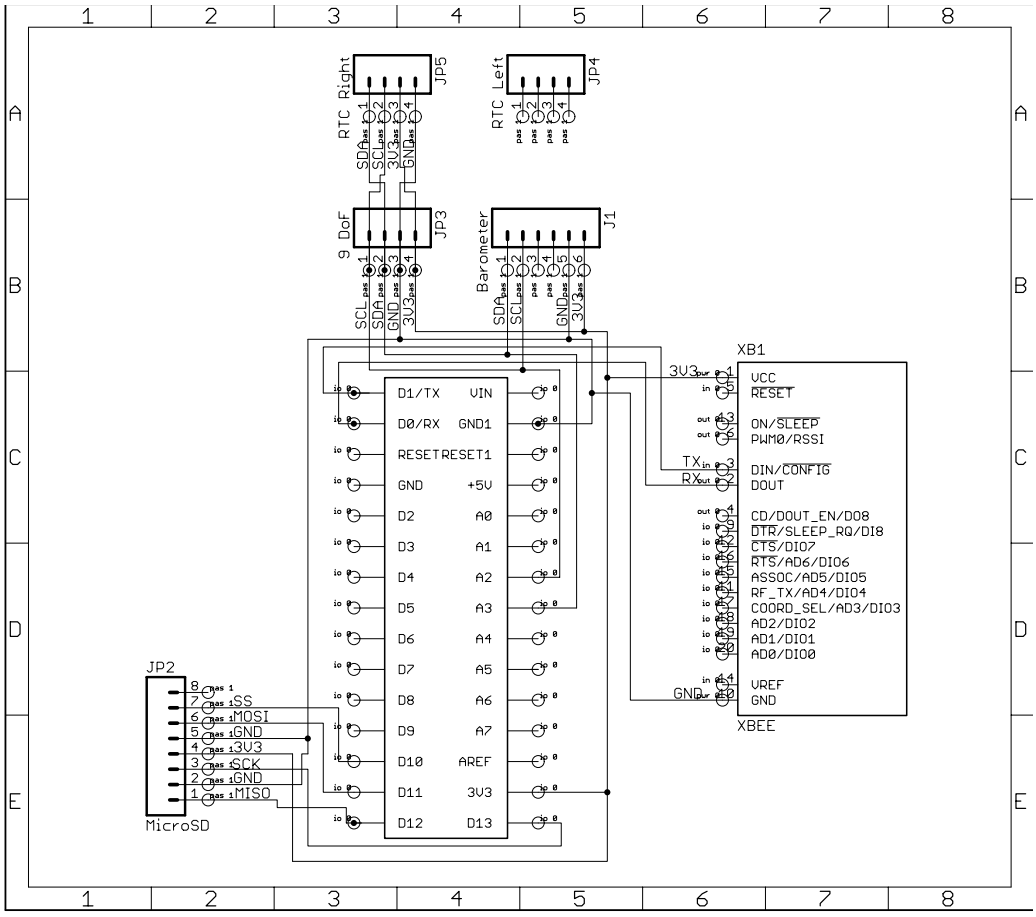


Figure A: Schematic of the OHDAD hardware

Appendix B

FLOW DIAGRAM OF THE OHDAD INITIALISATION PROCESS

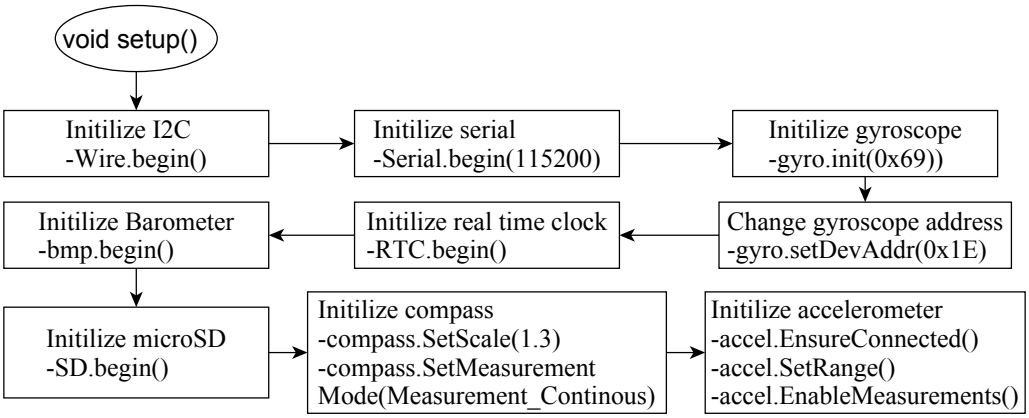


Figure B: Flow diagram to initialise the OHDAD

Appendix C

FLOW DIAGRAM OF THE SENSOR DATA COLLECTION PROCESS

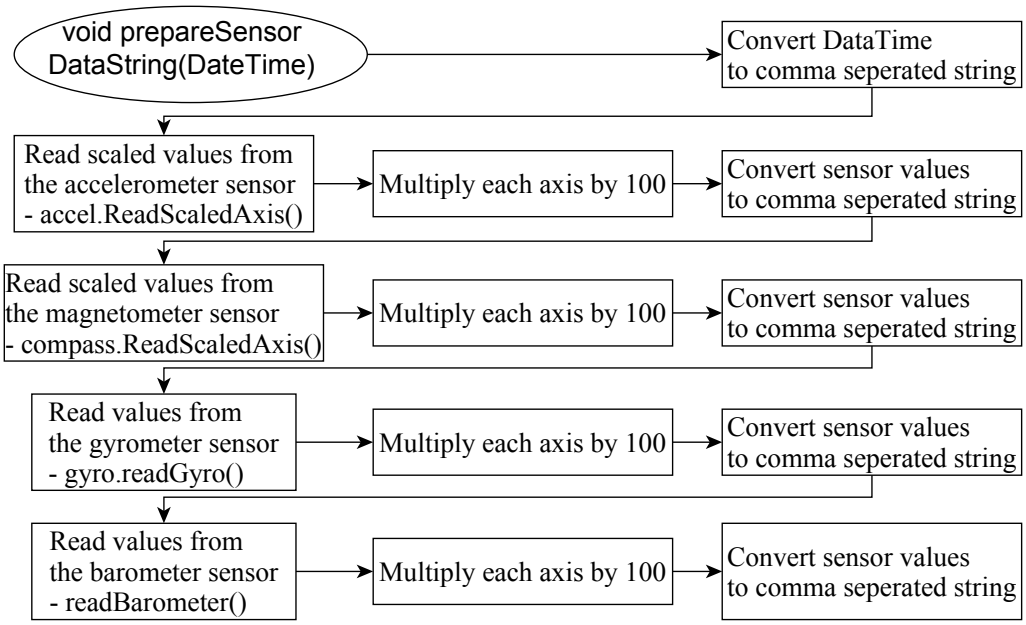


Figure C: Flow diagram to read the sensors attached to the OHDAD

Appendix D

FLOW DIAGRAM OF THE AIS DETECTOR COMPARISON

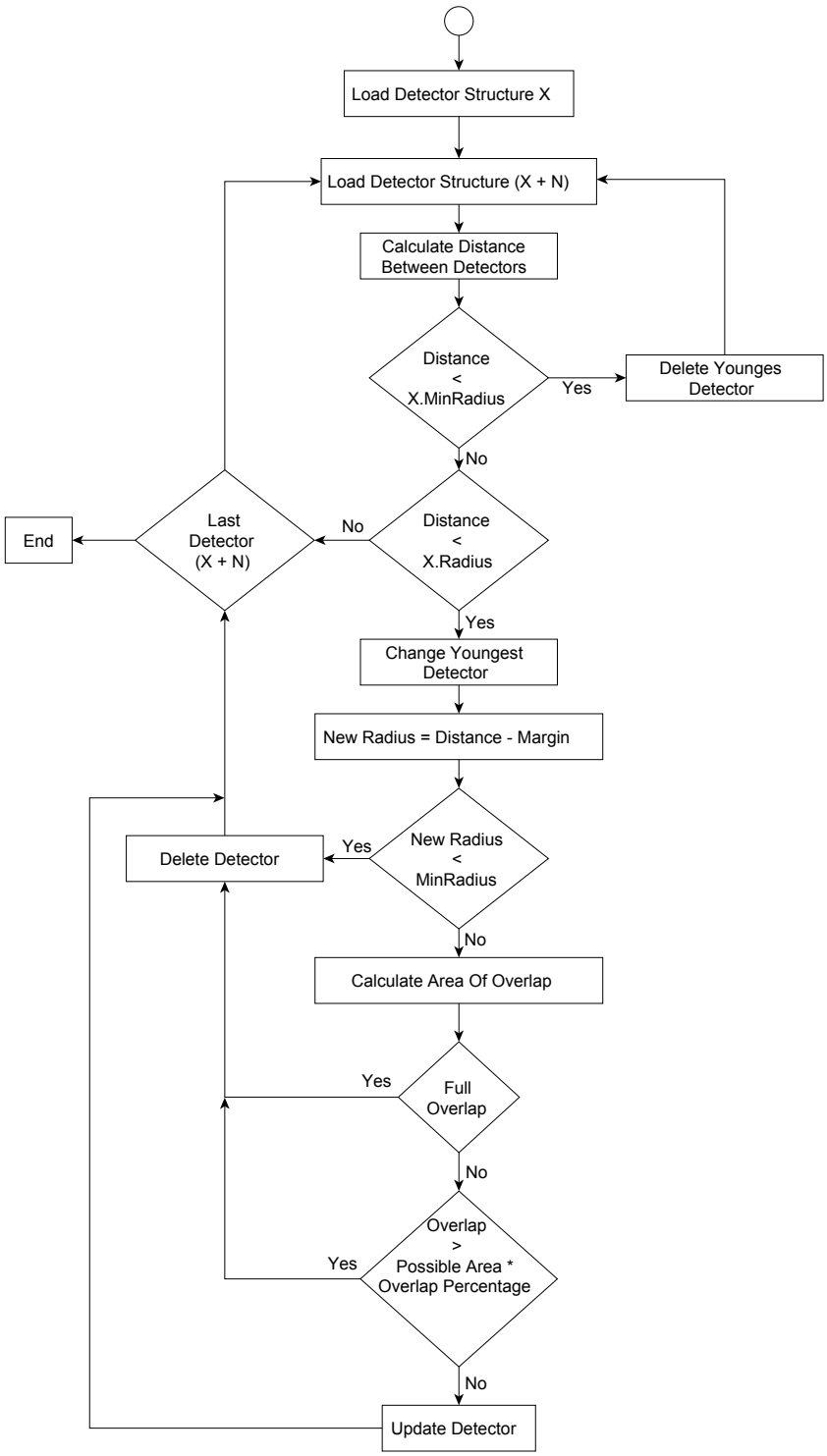


Figure C: Flow diagram to compare AIS detectors

Appendix E

FLOW DIAGRAM OF THE AIS ALARM RESET AND SEARCH SPACE OPTIMISATION

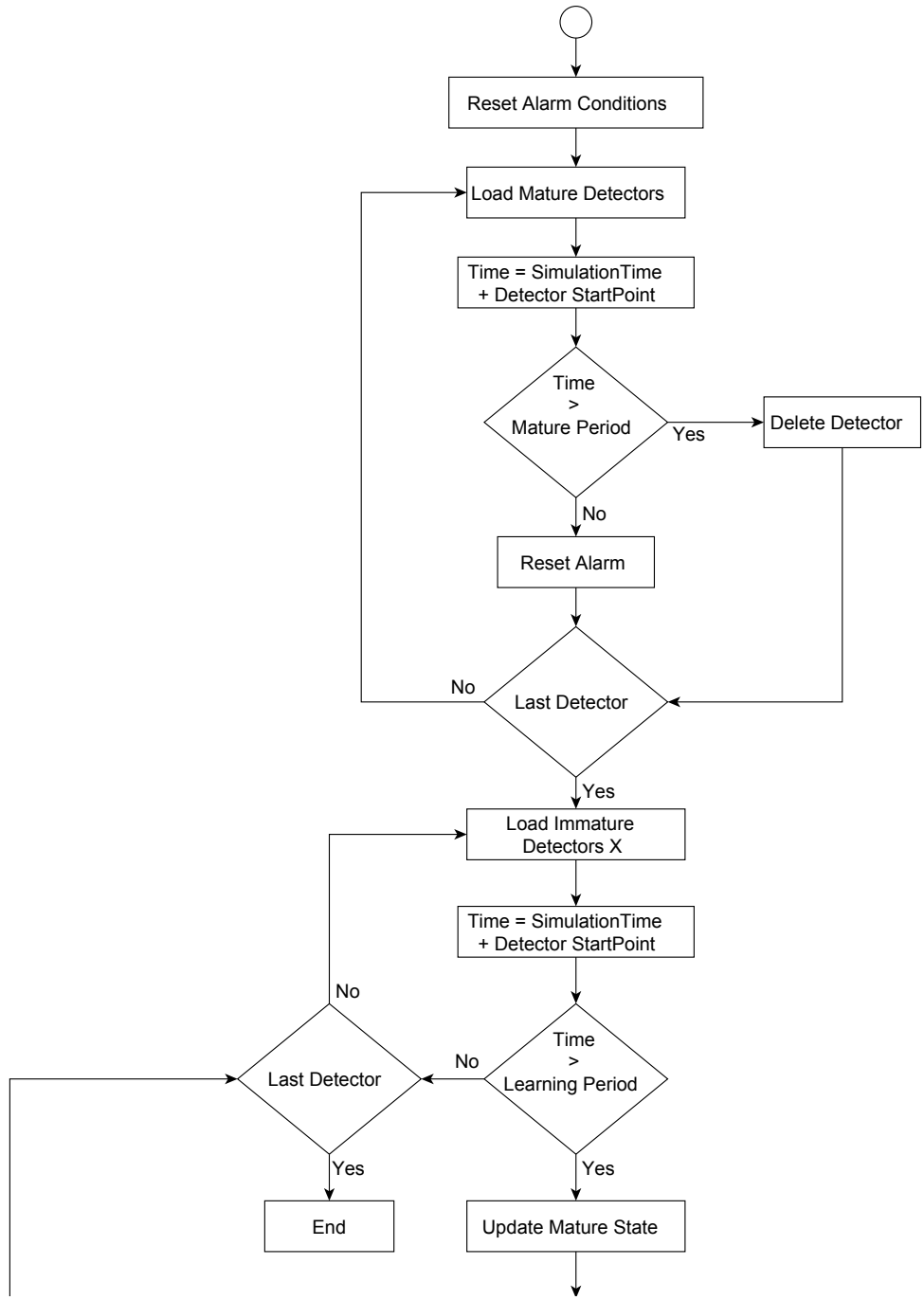


Figure E-1: Flow diagram of resetting detector alarm conditions and monitoring detector maturing phase

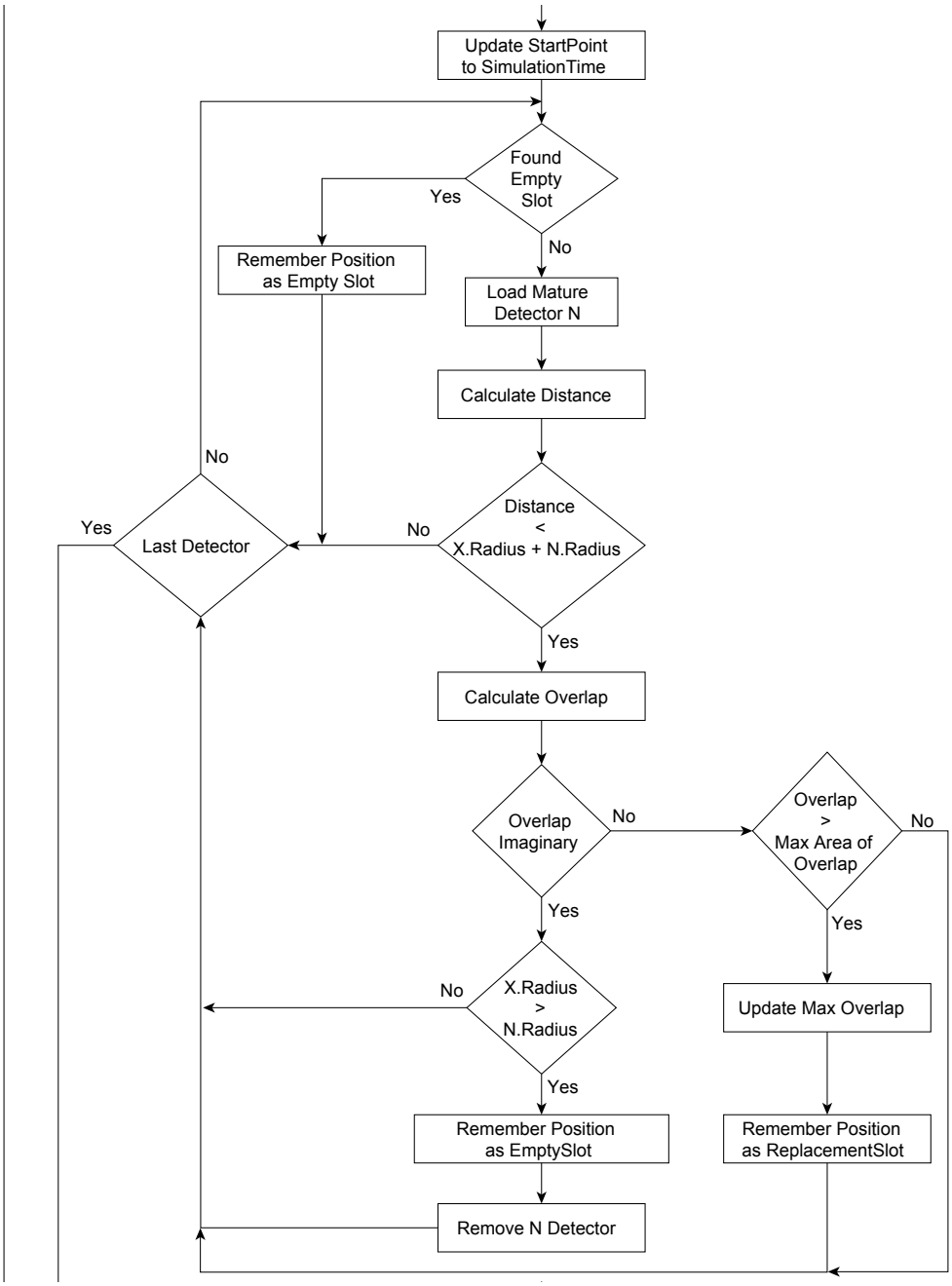


Figure E-2: Flow diagram of resetting detector alarm conditions and monitoring detector maturing phase

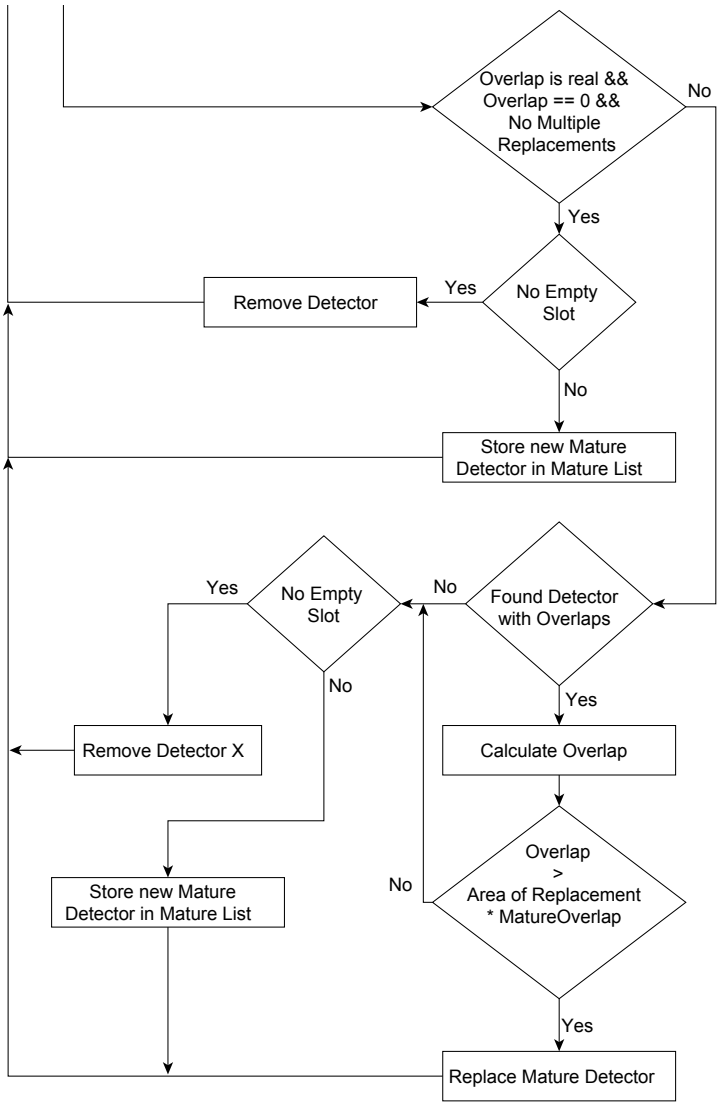


Figure E-3: Flow diagram of resetting detector alarm conditions and monitoring detector maturing phase

Appendix F

CALCULATION OF THE CIRCLE-CIRLE AND SPHERE-SPHERE INTERSECTION

Equation F.1 presents the calculation of the Circle-Circle Intersection, with $Distance$ being the distance between two detectors and R_1 and R_2 the respective affinity radii.

$Overlap = Part_1 + Part_2 + Part_3$	(F.1)
$Part_1 = R_1^2 * \text{acos}(\frac{Distance^2 + R_1^2 - R_2^2}{2 * Distance * R_1})$	(F.1.1)
$Part_2 = R_2^2 * \text{acos}(\frac{Distance^2 + R_2^2 - R_1^2}{2 * Distance * R_2})$	(F.1.2)
$Part_3 = \frac{1}{2}\sqrt{(Part_{3_1}) * (Part_{3_2}) * (Part_{3_3}) * (Part_{3_4})}$	(F.1.3)
$Part_{3_1} = -Distance + R_1 + R_2$	(F.1.3.1)
$Part_{3_2} = Distance + R_1 - R_2$	(F.1.3.2)
$Part_{3_3} = Distance - R_1 + R_2$	(F.1.3.3)
$Part_{3_4} = Distance + R_1 + R_2$	(F.1.3.4)

Equation F.2 presents the calculation of the Sphere-Sphere Intersection, with $Distance$ being the distance between two detectors and R_1 and R_2 the respective affinity radii.

$Overlap = \frac{Part_1 * Part_2}{Part_3}$	(F.2)
$Part_1 = \pi * (R_1 + R_2 - Distance)^2$	(F.2.1)
$Part_2 = Distance^2 + 2 * Distance * R_2 - 3\,R_2^2 + 2 * R_2 * R_1 + 6 * R_2 * R_1 - 3 * R_1^2$	(F.2.2)
$Part_3 = 12 * Distance$	(F.2.3)

Appendix G

DOMINATING PARAMETER COMBINATIONS FOR THE BAO ET AL. DATA SET

Table G-1: Dominant parameter combinations for the full data set

Bao Data Set					
Full Set					
SF	SM	WS	CM	CA	CL
10	FOSW25	1.5	NaiveBayes	82.28	0.355
20	FOSW25	1.5	NaiveBayes	83.33	0.361
10	FOSW50	2.5	NaiveBayes	83.79	0.361
30	FOSW90	0.5	NaiveBayes	87.00	0.367
10	FOSW25	2.5	KNN	90.34	0.376
20	FOSW25	1.5	KNN	92.89	0.380
20	FOSW50	2.5	KNN	93.13	0.381
20	FOSW50	3	KNN	93.30	0.385
30	FOSW25	1.5	KNN	93.51	0.391
30	FOSW50	2	KNN	93.86	0.394
30	FOSW50	2.5	KNN	94.88	0.396
20	FOSW75	4	KNN	95.44	0.405
20	FOSW75	4.5	KNN	96.73	0.406
10	FOSW90	11	KNN	97.34	0.416
20	FOSW90	12	KNN	98.27	0.442
20	FOSW90	11	KNN	98.32	0.448
30	FOSW90	3.5	KNN	98.48	0.452
30	FOSW90	6.5	KNN	98.74	0.494
30	FOSW90	5.5	KNN	98.75	0.489
30	FOSW90	7.5	KNN	98.85	0.537

Table G-2: Dominant parameter combinations for the 10 Hz sampling frequency limited data set

Bao Data Set					
10Hz Limited					
SF	SM	WS	CM	CA	CL
10	FOSW25	1.5	NaiveBayes	82.28	0.355
10	FOSW50	2.5	NaiveBayes	83.79	0.361
10	FOSW75	4.5	NaiveBayes	86.01	0.370
10	FOSW75	5	NaiveBayes	86.22	0.373
10	FOSW90	1.5	NaiveBayes	86.64	0.374
10	FOSW25	2.5	KNN	90.34	0.376
10	FOSW50	5.5	KNN	92.47	0.384
10	FOSW75	4.5	J48	93.04	0.400
10	FOSW75	5	J48	93.22	0.402
10	FOSW75	5	KNN	94.93	0.410
10	FOSW90	11	KNN	97.34	0.416
10	FOSW90	18	KNN	97.55	0.445
10	FOSW90	21	KNN	97.60	0.457
10	FOSW90	7.5	KNN	97.75	0.464

Appendix H

DOMINATING PARAMETER COMBINATIONS FOR THE OPPORTUNITY DATA SET

Table H-1: Dominant parameter combinations for the full data set

Opportunity Data Set					
Full Set					
SF	SM	WS	CM	CA	CL
20	FOSW25	0.5	NaiveBayes	64.18	0.376
20	FNSW	0.5	NaiveBayes	64.40	0.380
20	FOSW25	0.5	KNN	77.19	0.381
10	FOSW50	0.5	KNN	77.66	0.388
10	FOSW75	0.5	KNN	84.86	0.395
30	FOSW75	1	KNN	85.05	0.421
10	FOSW90	1	KNN	88.72	0.425
10	FOSW90	1.5	KNN	92.20	0.431
30	FOSW90	0.5	KNN	94.31	0.443

Table H-2: Dominant parameter combinations for the 10 Hz sampling frequency limited data set

Opportunity Data Set					
10Hz					
SF	SM	WS	CM	CA	CL
10	FOSW90	0.5	NaiveBayes	63.22	0.384
10	FOSW50	0.5	KNN	77.66	0.388
10	FOSW75	0.5	KNN	84.86	0.395
10	FOSW90	1	KNN	88.72	0.425
10	FOSW90	1.5	KNN	92.20	0.431

Sensors 2014, *14*, 4239-4270; doi:10.3390/s140304239

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Sensor Data Acquisition and Processing Parameters for Human Activity Classification

Sebastian D. Bersch [1,*], Djamel Azzi [1], Rinat Khusainov [1], Ifeyinwa E. Achumba [1] and Jana Ries [2]

[1] School of Engineering, University of Portsmouth, Anglesea Building, Anglesea Road, Portsmouth PO1 3DJ, UK; E-Mails: Djamel.Azzi@port.ac.uk (D.A.); Rinat.Khusainov@port.ac.uk (R.K.); ifeyinwaeucharia@gmail.com (I.E.A.)

[2] Portsmouth Business School, University of Portsmouth, Richmond Building, Portland Street, Portsmouth PO1 3DE, UK; E-Mail: Jana.Ries@port.ac.uk

[*] Author to whom correspondence should be addressed; E-Mail: Sebastian.Bersch@port.ac.uk; Tel.: + 44-239-284-2580.

Received: 21 November 2013; in revised form: 14 January 2014 / Accepted: 21 February 2014 / Published: 4 March 2014

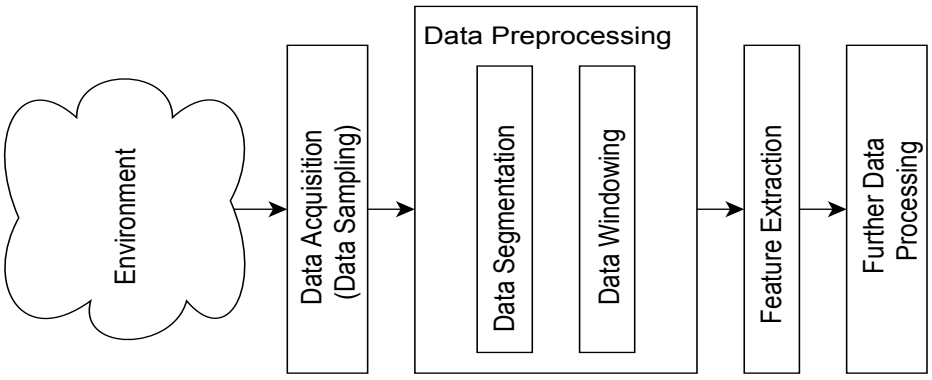
Abstract: It is known that parameter selection for data sampling frequency and segmentation techniques (including different methods and window sizes) has an impact on the classification accuracy. For Ambient Assisted Living (AAL), no clear information to select these parameters exists, hence a wide variety and inconsistency across today's literature is observed. This paper presents the empirical investigation of different data sampling rates, segmentation techniques and segmentation window sizes and their effect on the accuracy of Activity of Daily Living (ADL) event classification and computational load for two different accelerometer sensor datasets. The study is conducted using an ANalysis Of VAriance (ANOVA) based on 32 different window sizes, three different segmentation algorithm (with and without overlap, totaling in six different parameters) and six sampling frequencies for nine common classification algorithms. The classification accuracy is based on a feature vector consisting of Root Mean Square (RMS), Mean, Signal Magnitude Area (SMA), Signal Vector Magnitude (here SMV), Energy, Entropy, FFTPeak, Standard Deviation (STD). The results are presented alongside recommendations for the parameter selection on the basis of the best performing parameter combinations that are identified by means of the corresponding Pareto curve.

Keywords: Ambient Assisted Living (AAL); data acquisition; data sampling; event classification; optimization

1. Introduction

Ambient Assisted Living (AAL) is currently on the research agenda of many stakeholders worldwide, especially in Western countries, driven mainly by the needs of an aging population and in an attempt to address the demands of care and intervention for the elderly and those who require care. The main areas of interest in Assisted Living (AL) include fall prevention, promotion of independence, as well as ambulation and Activity of Daily Living (ADL) monitoring (for fall detection, activity recognition and classification). The timeliness and accuracy of the classification of ADL activities could have severe consequences if inadequate, especially in the case of an emergency event such as a fall and are therefore essential to provide the elderly with a sense of security and confidence [1,2]. Furthermore, reasonable levels of ADL facilitate the promotion of independence, hence the need for ambulation and ADL monitoring. Consequently, automated monitoring of subjects living independently in their homes, using wearable and off-body sensor-based devices, has been the subject of numerous research studies. While the literature highlights a great number of research areas for assisted living, such as sensor designs, placement of monitoring devices, novel monitoring techniques, fall detection and ADL data collection and classification methods, it fails to clarify some of the underlying and fundamental aspects of data collection in this field such as data acquisition and pre-processing (outlined in Figure 1, presenting standard prerequisites before ADL classification can take place).

Figure 1. Pre-steps before ADL.



Falls and ADL events are generally classified based on the features extracted from segments of the monitoring sensor data and have therefore a significant role in the accuracy of event classification [3]. Even though researchers are aware of the importance of sampling frequency; segmentation method; and window size with respect to feature extraction, the issue is not addressed in the reviewed studies with no clear explanation or justification given for the parameter selection. Furthermore, researchers tend to ignore the required Computational Load (CL) for data classification, which is of particular interest once data classification takes place on an embedded system for real time ADL recognition.

The literature review showed that there is no consensus in the selection of parameter combinations which once chosen, are seldom varied by researchers to improve classification results. Therefore, the work described in this paper empirically investigates the influence of sampling frequency (SF), segmentation method (SM), and windows size (WS) on the classification accuracy (CA) and computational load (CL) using two independent datasets (from Bao *et al.* and Roggen *et al.*). The work presented here tests eight commonly used features that are obtained from the accelerometer sensor data to determine CA and CL. The input information for the classifier are Root Mean Square (RMS), Mean, Signal Magnitude Area (SMA), Signal Vector Magnitude (here SMV), Energy, Entropy, FFTPeak, and Standard Deviation (STD). The results have been analysed using an ANalysis Of VAriances (ANOVA) to reveal the influence of the parameter combinations on the CA and CL. This is followed by an approach to recommend the parameter combinations that achieve the best CA disregarding CL and vice versa. Other parameter combinations may represent interesting trade-off points between these two preferences. This may be required in situations where time and hardware resources are limited. The authors aim to provide a more informed approach to parameter selection for event classification (with respect to the investigated ADLs) in the area of AAL.

Section 2 will highlight existing literature to outline the inconsistency and insufficient justification for parameter selection in ADL classification. This section also presents the process of data acquisition and introduces different segmentation techniques. Section 3 describes the investigation procedure. Section 4 presents the experimental results with a recommendation for parameter combinations, and Sections 5 and 6 present the discussion of results and conclusion.

2. Divergence in the Parameter Selection

2.1. Sampling Rate

The acquisition of data is one of the most critical steps in event classification as re-running experiments with test subjects is not always possible. Undersampling leads to loss of information and oversampling can result in information buried in unwanted noise. In the latter case, longer computational time is needed for analysis as more data needs to be processed. The minimum sampling rate $f_{sampling}$ is dependent on the maximum frequency contained in the data signal f_{max} (the sampling theorem) [4]. In the area of AAL, a review of the literature has not uncovered a typical sampling frequency.

The highest sampling rate for AAL that the authors found during their research is 512 Hz by [5] followed by the works of [6] where the authors use a sampling rate of 256 Hz to collect accelerometer data. [7] use a two-axis accelerometer and a sampling frequency of 76.25 Hz, which is less than 1/3 of [6] sampling rate. [8] choose $f_{sampling}$ to be 64 Hz. The authors acknowledge the high frequency sampling rate used by [6] however they reduced the sampling frequency on the bases that lower values are more feasible with off-the-shelf activity monitors. They further mention the work of [9], who sample accelerometer data at 50 Hz, therefore resampling their own data at the same frequency as well. Overall the literature highlights that values around 50 Hz are one of the more common sampling rates. [10] use 52 Hz, [11] use 50 Hz to sample their tri-axial accelerometer, while [12] and [13] also report a 50 Hz sampling rate for an eWatch with two-axis accelerometer and a light sensor. To the authors' best knowledge, [13] are the only ones that tested different sampling frequencies (from 1 to 30 Hz)

for the sensor data. The outcome highlights that the recognition of ADLs improves with higher sampling rates but only marginally improves with sampling rates above 20 Hz. In [14] the authors demonstrate that 98% of the FFT spectrum amplitude is contained below 10 Hz, and 99% below 15 Hz. This corresponds to the findings of [15] who state that a sampling frequency of 20 Hz is sufficient to successfully classify ADLs. The lowest sampling rate that the authors found in the literature is 5 Hz by [16].

2.2. Data Preprocessing Techniques

2.2.1. Segmentation Method

One of the challenges of data pre-processing following acquisition consists in deciding which points to actually use in the live stream of data. Several different segmentation methods exist to divide a larger data stream into smaller fit for processing chunks. The selection of the right segmentation technique is crucial, as it immediately impacts on the extracted features used for the ADL classification and the resulting classification accuracy. Therefore even the best classifier performance will be weak when the extracted features are non-differentiable [3]. Furthermore, the segmentation techniques can also have an impact on the real time capabilities as complex segmentation methods can increase CL but might result in improved classification accuracy. Moreover, the segmentation method also dictates how often features need to be extracted and classification algorithms need to be executed. Literature has highlighted several different segmentation techniques used in various research projects, such as: Fixed-size Non-overlapping Sliding Window (FNSW) [3,17], Fixed-size Overlapping Sliding Window (FOSW) [3,17], Top-Down (ToD) [17], Bottom-Up (BUp) [17], Sliding Window And Bottom-up (SWAB) [17], Symbolic Aggregate approXimation (SAX) [3], String Matching (SM) [3], Reference-based Windowing (RbW) [18], Dynamic Windowing (DWin) [19] and Variable-size Sliding Window (VSW) [20]. The significant difference in these techniques resides in their online and offline capabilities. The meaning of an online technique is that the data can be segmented before the complete data is collected, while offline methods require the entire dataset first. For real time applications, only online techniques are of interest. [17] note that online algorithms can produce very poor approximations of data under certain conditions but have a relatively good performance on noisy data. However, the authors also highlight that the FOSW segmentation algorithm is of particular interest in medical research, e.g., patient monitoring as the algorithm is simple and intuitive for researchers to understand. As part of this paper the algorithms investigated should be fairly simple to understand and online capable (FNSW, FOSW, and SWAB).

2.2.2. Window Size

Researchers who use fixed size window segmentation methods apply inconsistent window sizes. [10] use especially short windows of 1 s. [8] report to use a 2 s window based on their short ADLs in their research and because they achieve only a minimal gain in classification accuracies with features from a 3 s window. Further examples for short windows are [21] and [9], with 2 s and 2.56 s respectively. [13] extract features from a 4 s buffer, [12] use 5 s in their research, and [7] report a window length of 6.7 s. While these researchers are using short window sizes, [20] describe the usage of a 60 s windows in the work of [22] and 74 s in [23]. Furthermore, [20] introduce possible

modification to the fixed size window methods. The authors suggest in their work to dynamically adjust the window size based on special events in the sensor data as different ADLs have different time frames. They raise the point that longer window sizes can cover more than one ADL while a small window could split an activity, which both leads to suboptimal information for an activity classifications algorithm.

Table 1. Inconsistency in sampling rates and segmentation windows for AAL.

Authors	Sampling Frequency [Hz]	Segmentation Window [s]	Segmentation Method	Testsubject Information	ADLs
Huynh [5]	512	0.25, 0.5, 1, 2, 4	FNSW, FOSW 50%, FOSW 75%, FOSW 80.5%, FOSW 93.75%		Walking, Standing, Jogging, Skipping, Hopping, Riding Bus
Sekine [6]	256			Subjects 11; Age 69.3 ± 5.6 years; Height 1.54 ± 0.078 m; Weight 50.4 ± 9.6 kg	Walking
Bao [7]	76.25	6.7	FOSW 50%	Subjects: 13 male, 7 female; Age 17–48 years	Walking, Sitting & Relaxing, Standing Stil, Watching TV, Running, Stretching, Scrubbing, Folding Laundry, Brushing teeth, Riding Elevator, Walking Carrying items, Working on Computer, Eating or Driniking, Reading, Bicycling, Strength Training, Vacuuming, Lying Down & Relaxing, Climbing Stairs, Riding Escalator
Preece [8]	64	2 and 3	FOSW 50%	Subjects: 10 male, 10 female; Age 31 ± 7 years; Height 1.71 ± 0.07 m; Weight 68 ± 10 kg; BMI 24 ± 3	Walking, Walking Upstairs, Walking Downstairs, Hopping on Left Leg, Hopping on Right Leg, Jumping
Wang [9]	50	2.56	FOSW 50%	Subjects: 39 male, 12 female; Age 21–64 years; Height 1.53–188 m; Weight 42–94 kg	Walking, Walking Slope Up, Walking Slope Down, Walking Stairs Up, Walking Stairs Down
Casale [10]	52	1	FOSW 50%	Subjects: 11 male, 3 female	Walking Stairs Up, Walking Stairs Down, Walking, Talking, Staying Standing, Working at Computer
Ravi [11]	50	5.12	FOSW 50%	Subjects 2	Standing, Walking, Running, Walking Stairs Up, Walking Stairs Down, Situps, Vacuuming, Brushing Teeth
Pärkkä [12]	50	5		Subjects 7; median (range); 27 years (4–37); Height 180 (92–187)	Lying, Sitting, Standing, Walking, Bicycling, Running
Maurer [13]	50	4	FOSW 92%	Subjects 6	Sitting, Standing, Walking, Walking Stairs Up, Walking Stairs Down, Running
Antonsson [14]	1–30			Subjects 12	Walking (Gait)
Bouten [15]	20			Subjects: 13 male; Age 27 ± 4 years; Height 1.83 ± 0.07 m; Weight 77 ± 12 kg	Sedentary Activities, Household Activities, Walking
Gjoreski [16]	5	1.4			Standing, Lying, Sitting, On all fours, Sitting on the Ground, Going Down, Standing Up

Table 1. *Cont.*

Authors	Sampling Frequency [Hz]	Segmentation Window [s]	Segmentation Method	Testsubject Information	ADLs
Nyan [21]	256	2		Subjects 22; Age 20–45 years; Height 1.67–1.94 m; Weight 45–93 kg	Walking, Walking Upstairs, Walking Downstairs
Kasteren [22]		60	FNSW	Subject 1	Leaving House, Toileting, Showering, Sleeping, Breakfast, Dinner, Drink
Patterson [23]		74		Subject 1	Using Bathroom, Making Oatmeal, Making Soft-Boiled Eggs, Preparing Orange-Juice, Making Coffee, Making Tea, Making or Answering a Phone Call, Taking out the Trash, Setting the Table, Eating Breakfast, Clearing Table
Pietka [3]			FNSW, FOSW, SAX, SM		
Keogh [17]			FNSW, FOSW, Bup, SWAB		
Chu [18]			RbW		
Kozina [19]			Dwin		
Ortiz Laguna [20]			VSW		

The work of [16] has a similar point, indicating that to achieve good classification accuracy, different sensor features should be extracted using varying window sizes. These methods lead to complex monitoring systems if several ADLs need to be classified. Each feature window could yield different ADL classification results, which would then require a voting system to predict the correct ADL from the list of possible activities. Section 2 highlighted the divergence in parameter selection in the literature covering ADL event classification. Table 1 represents a summary of the different combinations discovered. The section above pointed out problems that are introduced when the wrong sampling frequency (over/undersampling) is used for the data acquisition. It also showed that researchers in the field are not in agreement over which sampling rate to use. The section also showed the use of various window sizes covering a wide range of values. Most studies base their parameter combinations on past experiments, hardware limitations, or do not state a specific reason. It was also found that possible CA or CL improvements based on different combinations are neither investigated nor mentioned. This inconsistency in parameter selection is the fundament for the study presented here for a more informed decision on parameter selection.

3. Investigation Procedure

This work presented here was based on two different datasets from the literature. The first dataset contains two-axis accelerometer data collected by [7]. The data contains 20 different participants (13 males and seven females with a mean age of 21.8 years (±6.59 years SD)), which were recruited at the MIT with the help of posters. The experiment required the test subjects to execute several different ADLs

under laboratory conditions without any supervision or guidance. Sensor data was collected simultaneously at five different positions (ankle, thigh, wrist, hip, upper arm), with a sampling frequency of 76.25 Hz. From the five sensor positions, the data of the hip sensor from all twenty different participants were used with the focus on ADLs such as, walking, sitting, walking and carrying an item, standing still, lying down, and climbing stairs. The second dataset is the Opportunity dataset collected as part of a European Funded project by [24]. The dataset is not limited to just body-worn accelerometer data. The complete set includes a total of 10 different sensor types, such as microphone, magnetometer, UWB localization, RFID, *etc.*, totaling a collection of 72 sensors. Data was recorded with 12 test subjects, which are not further specified. Of these 12 subjects only three subjects are labeled and available in the UCI Machine Learning Database. The labeled locomotion activities that were used from this dataset are stand, walk, sit, and lie. As both datasets should be similar to allow for comparable result, the Opportunity dataset was limited to the accelerometer (sampled at 64 Hz) placed at the subject's hip and limited to the *x* and *y* axis. Using Matlab [25], the sensor data was resampled, segmented and the data features extracted, while the Weka software package [26] provided the implementation of the classification algorithms.

3.1. Resampling

Section 2 showed that sampling rates vary greatly throughout the literature; it also indicated a high use of sampling rates around 50 Hz even though the work of Maurer *et al.* argues that sampling rates above 20 Hz only marginally improve classification accuracy [13]. Therefore the complete data set was resampled using Matlab at six different sampling rates in the range of 10 to 60 Hz in 10 Hz steps. Intermediate steps were ignored for the benefit of faster experiments, as well as the authors' belief that the omission of intermediate steps would not cause loss of generality of the results. Additionally, sampling rates above 60 Hz were excluded, as the authors concur with [8], who stated that higher sampling rates are harder to achieve with off-the-shelve-components.

3.2. Data Segmentation

The work presented here focuses on three online segmentation techniques: FNSW, FOSW (with four different overlap percentages), and SWAB that were introduced in Section 2.2. As described above, the advantages of these algorithms are that they are online capable, therefore can be used while the data collection is in progress and are simple and intuitive so that they are easily understood.

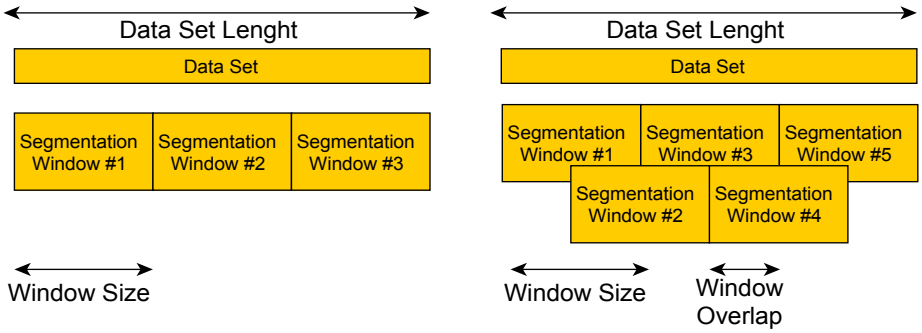
FNSW is a simple segmentation technique without any data overlap (see Figure 2a). The end point of segmentation window *N* is the starting point for window *N + 1*. It is therefore possible to exactly calculate the amount of windows generated for a given data set size with Equation (1):

$$Segmentation_{Windows} = \frac{S}{L_i} \quad (1)$$

where S is the total number of signal samples and $L_i = R_{Sampling} * Window_{Size}$, where $R_{Sampling}$ is the data resampling rate used (in the range of 10 to 60 Hz) and $Window_{Size}$ is the selected window size (in the range of 0.5 to 24 s). One disadvantage of this technique is that data associated with a particular feature (e.g., fall) can be split between windows. A FNSW sliding technique that is not

covered in this paper is to leave a gap between adjacent windows, as this would result in uncovered sensor data and therefore could miss important information.

Figure 2. Explanation of segmentation method. (a) FNSW; (b) FOSW with 50% overlap.



The FOSW segmentation technique is based on FNSW but includes data overlap (see Figure 2b showing FOSW with an overlap of 50%). Depending on the percentage overlap, more or less data overlaps from window *N* into *N + 1*. This is also referred to as a window shift. A 0% overlap corresponds to the FNSW segmentation method, while an overlap of 100% would yield to a static window as it would not be shifted and the data would always be segmented at the exact same point. Therefore, the requirement for FOSW is to move with at least one data point per segmentation. The number of segmentation windows generated can be calculated using Equation (2):

$$Segmentation_{Windows} = \frac{S}{L_i - L_i * \frac{P_j}{100}} \quad (2)$$

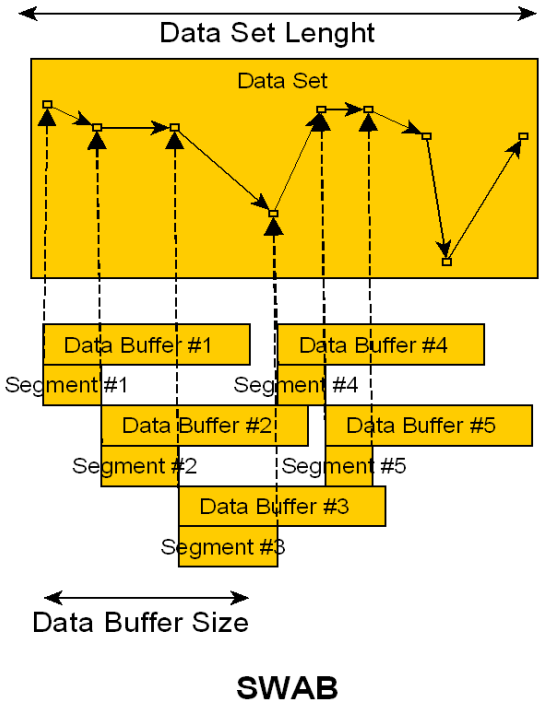
with P_j being one of the following percentage overlap values used for this research: 25, 50, 75, and 90.

SWAB is the third segmentation technique used as part of the study presented here and was designed by [17]. It is a combination of the Sliding Window and Bottom-up approach. The process is visualized in Figure 3. The algorithm has a fixed size data buffer that is used for the Bottom Up approximation, which joins the smallest approximation segments until a stopping condition is met. Once the approximation for the window is complete the data buffer shifts by the first segment (here identified as Segment #1 in the illustration) and the process is repeated for the new buffer window. Each segment is used for feature extraction. As the data shift is dependent on the dataset and its approximation, it is not possible to estimate the amount of segmentation windows generated by the algorithm. The implementation is more complex compared to the FNSW and FOSW methods described above and therefore an increased CL is expected while the CA gain is uncertain even though literature suggests improved results.

As highlighted earlier in Section 2.2, there is no clear recommendation in the published literature on the selection of the window size used for the data segmentation. The authors therefore tested a range of 32 different sizes in the range of 0.5 to 24 s. In the area of 0.5 to 8 s, the size is increased in 0.5 s steps, while thereafter the step size is increased to 1 s. The 0.5 s step size was increased after 8 s because the ADLs under investigation have only a short time frame and computational load was reduced for the experiment. Even though literature showed the use of longer window sizes the aim is to only include

single ADLs in each window to achieve the best classification results. The authors' initial research [27] supports this idea as it indicates a decrease in accuracy for window sizes above 8 s. Furthermore, it is the authors' belief that most ADLs will only take a short amount of time and a maximum of 24 s should be sufficient to include at least two ADLs.

Figure 3. Explanation of segmentation method SWAB.



3.3. Data Feature Selection

The following eight metrics are quite common in the area of ADL classification and therefore used to retrieve the different features of the accelerometer sensor data in this research: Root Mean Square (RMS), Mean, Signal Magnitude Area (SMA), Signal Vector Magnitude (here SMV), Energy, Entropy, FFTPeak, Standard Deviation (STD). These metrics and their significance are discussed below as each individual metric has its own influence in the research field.

RMS has been used to distinguish walking patterns [6] as well as being an input to classifiers for activity recognition [16,28]. The RMS value is calculated using Equation (3):

$$RMS_{x-Axis} = \sqrt{\frac{\sum_{i=1}^{n} {x_i}^2}{n}} \quad (3)$$

The Mean metric (see Equation (4)) has been used to: recognize sitting and standing [28,29]; it discriminates between periods of activity and rest [30]; and as an input to classifiers such as Decision Table, KNN, J48, Naïve Bayes, Random Forest, Hidden Markov Model (HMM) [7,16,31,32]:

$$Mean_{X-Axis} = \frac{\sum_{i=1}^{n} x_i}{n} \quad (4)$$

The next metric, SMA is used to distinguish between periods of activity and rest in order to identify when the subject is mobilizing and undertaking activities, and when they are immobile [15,33,34]. Equation (5) implements SMA:

$$SMA = \frac{\sum_{i=1}^{n}(|\ x_i\ | + |\ y_i\ |)}{n} \tag{5}$$

SMV, normally referred to as Signal Vector Magnitude (SVM) but changed to SMV to avoid confusion with the SVM classifier used, indicates the degree of movement intensity and is an essential metric in fall detection [33,34]. The SMV value is calculated using Equation (6):

$$SMV = \frac{\sum_{i=1}^{n}\sqrt{{x_i}^2 + {y_i}^2}}{n} \tag{6}$$

Two additional metrics used in this research are Energy and Entropy, which discriminate between types of ADL such as walking, standing still, running, sitting and relaxing [7,35]. The calculation of the Energy value is based on Equation (7) and the Entropy is calculated using the Matlab function from [36]:

$$Energy_{x-Axis} = \frac{\sum_{i=1}^{n} FFT_{x_i}^2}{n} - Mean_{X-Axis} \tag{7}$$

Another feature that was extracted from the accelerometer data stream is the FFTPeak for each axis. The metric has been used for activity recognition [5,12,35]. The FFTPeak algorithm was based on the Matlab Example found at [37].

The last metric used is Standard Deviation (STD), which has been extensively used for activity recognition [29]; and as an input to classifiers, such as J48, Random Forest and Artificial Neural Networks (ANN) [16,38]. Equation (8) describes the calculation:

$$STD_{X-Axis} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - Mean_{X-Axis})^2}{n}} \tag{8}$$

3.4. Classifier Selection

The software tool Weka implements the algorithms of several different classifiers from which nine were selected based on literature to verify the effects of changes in the described parameters above. [12] points out that common algorithms for activity classification are Support Vector Machines (SVM), Decision Trees, and Bayesian classifiers. The work of [13] included the use of Decision Tress and Naïve Bayes classifier. [10] based their research on Decision Trees, Bagging of 10 Decision Trees, AdaBoost using Decision Trees as base classifiers and a Random Forest of 10 Decision Trees. Additionally, in the work of [16] the authors compare J48 (Decision Trees) and Random Forest. This investigation therefore included the following classifiers: Naïve Bayes, SMO (based on SVM), KNN, KStar, MultiClassClassifier, Bagging, Decision Table, J48, and Random Forest. All algorithms were tested using Weka's standard configuration and a 10-fold cross validation. An additional classifier fine-tuning is a research field in its own and therefore not discussed here. The use of different classification methods has enabled the authors to verify the impact of the sampling rate, segmentation method, and window size on the classification accuracy over a wide field of algorithms used in AAL.

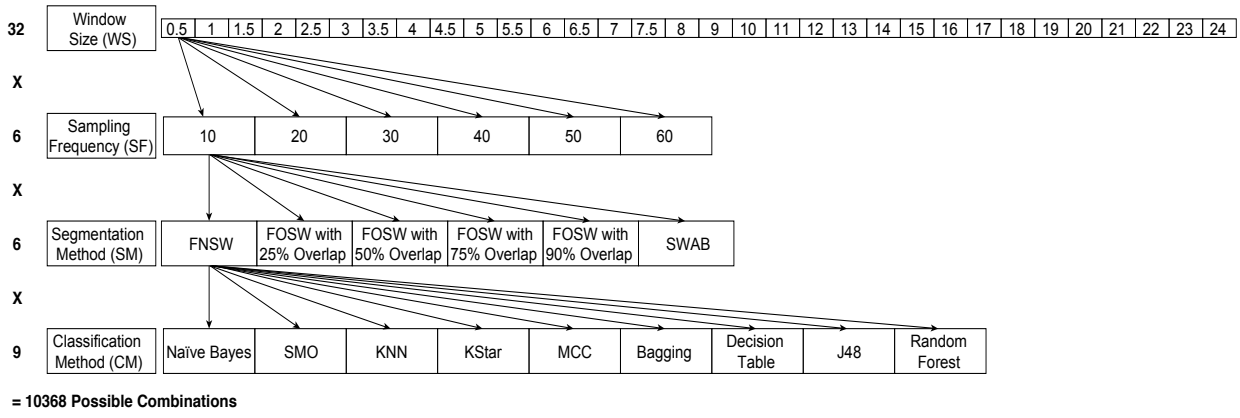
4. Experimental Results

This section will be split in three subsections to focus on different aspects of the parameter selection problem introduced by the variation in classification methods (CM), sampling frequency (SF), segmentation methods (SM) and window size (WS). The first two analyses use an ANalysis Of VAriance (ANOVA) to investigate the impact of the identified parameters on the two dependent variables classification accuracy (CA, as described in Section 4.1) and computational load (CL, as described in Section 4.2). The analyses have been conducted in SPSS [39]. In the third subsection the authors present the results of an investigation into the effects of various input parameter combinations on CA and CL with a view to enable optimum parameter selection based on the identification of the corresponding Pareto curve (see Section 4.3).

Figure 4 below shows the different levels of the four parameters CM, SF, SM, and WS; this has resulted in 32 different window sizes, three segmentation methods with different parameters (resulting in six SM levels) and six sampling frequencies for each of the nine different classification algorithms. This results in 10,368 different parameter levels for each of the 23 test subjects for the analysis of variance (ANOVA) in SPSS. The performance measures used are classification accuracy (CA) and computational load (CL), as explained in Section 4.2. Next to classification accuracy, literature also highlighted the use of precision, recall, and f-measure. In [40] the author argues that precision, recall, and f-measure are especially useful for highly imbalanced datasets. For example, when faced with a two-class problem with a split of 98% (majority) and 2% (minority), just guessing the majority class will achieve an accuracy of 98%. If the detection of the minority class, say, representing rare and infrequent events (e.g., falls), is important, an accuracy of 98% would be misleading in terms of the performance of the classifier. The datasets used for this research include six and four, respectively, different activity classes, which are roughly equally distributed and are equally important to classify. It is therefore adequate to use the accuracy of the classifier instead of f-measure:

$$Accuracy = \frac{Correct\ Classified\ Instances}{Total\ Instances} * 100 \quad (9)$$

Figure 4. Parameter combinations for each classifier.



4.1. Statistical Analysis of Accuracy

This section reports on the impact of the variations of four input parameters (CM, SF, SM, and WS) on classification accuracy (output). During the initial analysis of the ANOVA output, the tested accuracy for the KStar algorithm showed a strong sensitivity to changes in the input parameters (SF, SM, and WS). Its influence on the analysis was significant and superimposed on the results. As a consequence the impact of certain input parameters appeared to be of significance for the classification accuracy, while overall the impact resulted from the sensitivity of the KStar classifier. Therefore the authors decided to exclude KStar in the analysis in order to avoid a misinterpretation of the overall impact of parameters on accuracy.

4.1.1. Dataset Bao *et al.*

The ANOVA results (presented in Table 2), excluding KStar, showed that 49% of the variations in the dependent variable (accuracy) are described by the four input parameters. This means, that other input parameters that were not tested in the scope of this experiment may have further influence on the accuracy. Such a result is not surprising, as the investigated problem is highly complex and it is understandable that factors such as the test subject itself and the recorded movement have also an impact on the resulting accuracy.

Table 2. ANOVA output for the CA as the dependent variable (Tests of Between-Subjects Effects. Dependent Variable: CA).

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	3,257,844 [a]	670	4,862	260	0.000
Intercept	1,503,713,645	1	1,503,713,645	80,275,936	0.000
SF	116,380	5	23,276	1,243	0.000
WS	216,554	31	6,986	373	0.000
SM	650,201	5	130,040	6,942	0.000
CM	1,961,904	7	280,272	14,962	0.000
SF * SM	5,924	25	237	13	0.000
SF * CM	7,591	35	217	12	0.000
SF * WS	36,024	155	232	12	0.000
SM * WS	60,930	155	393	21	0.000
WS * CM	92,164	217	425	23	0.000
SM * CM	110,091	35	3,145	168	0.000
Error	3,439,779	183,633	19		
Total	1,510,410,864	184,304			
Corrected Total	6,697,622	184,303			

[a] R Squared = 0.486 (Adjusted R Squared = 0.485).

The table shows the Sums of Squares (a measure for the average variability in the data), Degree of Freedom (df—scores that are free to vary once the mean of the set of scores is known), Mean Square (which is used to estimated the variance), F (F-Ratio represents the indicator for the significance on performance caused by the independent variables instead of chance), and Sig. (indicating the significance

level at which the main/two-way interaction effects are significant <0.05 or non-significant >0.05) for all main and two-way interaction effects. The two-way interaction effects outline a changing main effect of one factor for different levels of a second factor and are therefore of higher interest than the main effect alone, if they are identified as being significant. The Sig. column in Table 2 shows that each main and the six two-way interaction effects (SM and CM, WS and CM, SM and WS, SF and WS, SF and CM, SF and SM) have a significant impact on the accuracy. Furthermore, the Type III Sum of Squares can be used as an indication for the importance of the main and two-way interaction effects. For an easier overview the rows are already sorted and it is observed that CM is the most influential factor followed by SM, WS, and SF in decreasing order. The significance for the two-way interaction effects starts with SM and CM and is followed in decreasing order by WS and CM, SM and WS, SF and WS, SF and CM and SF and SM.

Figure 5 shows the interaction effect of changes in the segmentation methods on the classifier methods. The Naïve Bayes classifier shows only a minor improvement in accuracy, whilst the remaining seven classifiers show a substantial increase with an increased segmentation overlap for FOSW. Another visible effect is the good performance of classifiers with the SWAB segmentation method, mostly outperforming FOSW with 75% overlap. For Naïve Bayes, SWAB showed a significant decrease in performance, which results in an even lower CA when compared to FNSW.

Figure 5. Two-way interaction effect for SM and CM.

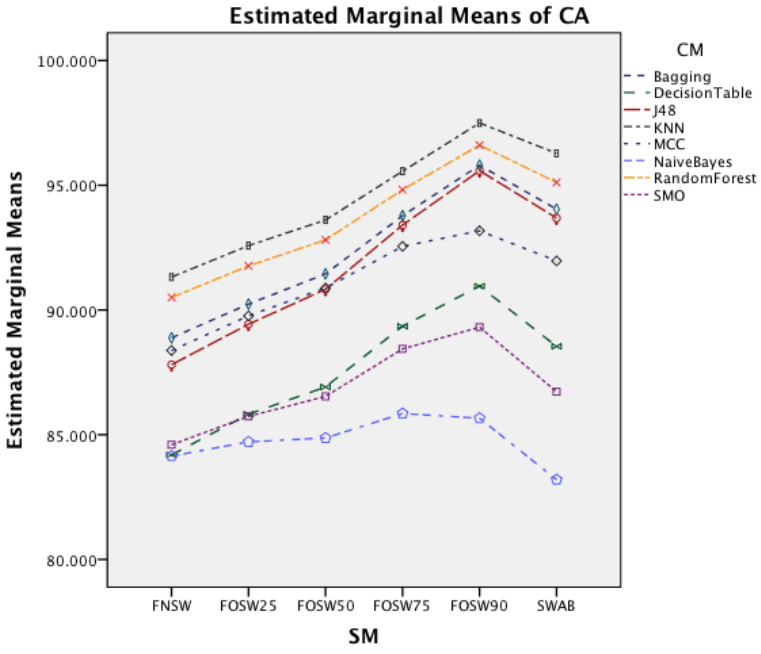


Figure 6 presents the effect of an increased window size on the different classifiers. Besides Naïve Bayes, all classifiers show a decrease in accuracy for window sizes below 7 s before stagnating and start to decrease after the window size increases above 9 s. Naïve Bayes is the only CM that actually improves CA with an increased window size.

The next effect investigated is the interaction between WS and SM in Figure 7. It is noticeable that an increased window size decreases the accuracy for each segmentation method. Furthermore, the effect reduces with an increased segmentation overlap, showing a less significant impact on FOSW

with 90% compared to FOSW with 25% overlap. SWAB follows the behavior of FOSW with 75% overlap with a decreased overall accuracy. The figure also shows that a window size of 6.5 to 11 s results in the best accuracy.

Figure 6. Two-way interaction effect for WS and CM.

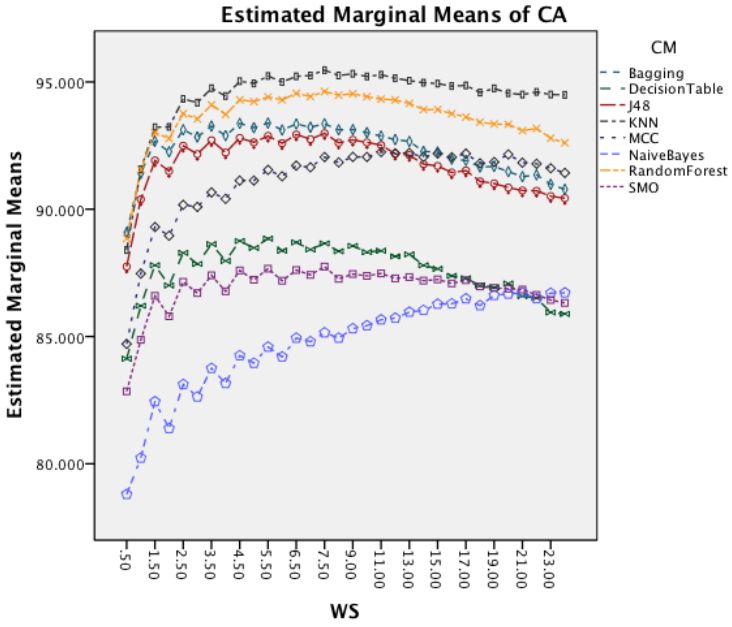


Figure 7. Two-way interaction effect for WS and SM.

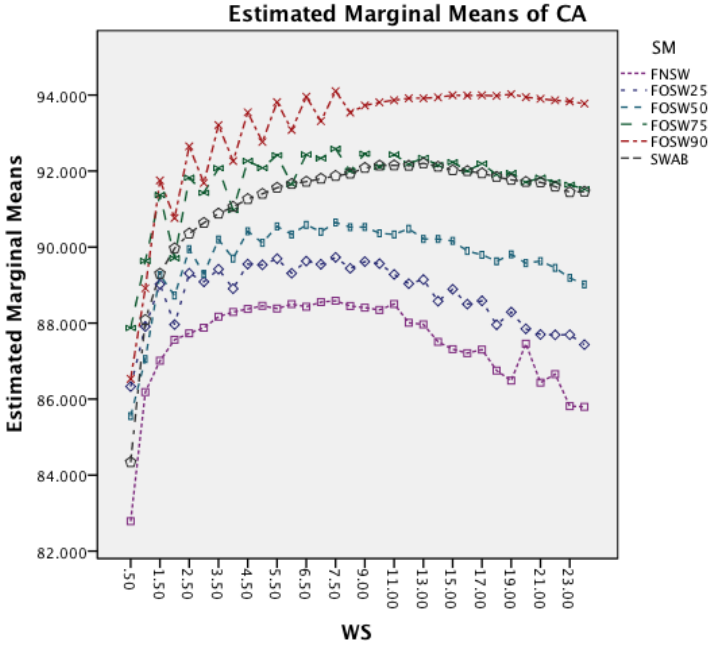
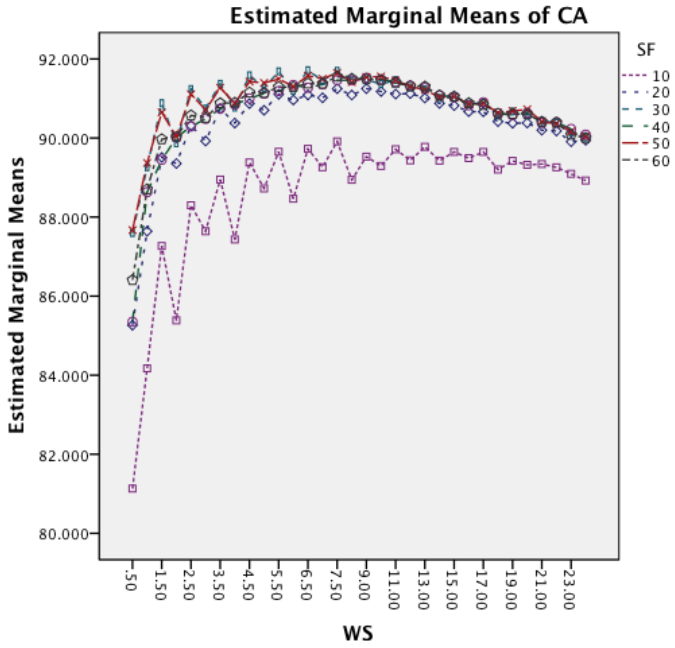


Figure 8 shows the interaction effect between WS and SF. The effect of an increased window size is similar for all six sampling frequencies, while the effect is reduced for a sampling frequency of 10 Hz for longer window sizes. The graph also highlights that higher frequencies achieve the best accuracy for shorter window sizes, while the 10 Hz sampling frequency requires a slightly larger window.

Figure 8. Two-way interaction effect for WS and SF.



The last effect under investigation is SF and CM. The graph is not presented, as there is no interaction effect between the different classifiers. The only effect that exists is a minor improvement of accuracy for a change of sampling frequency from 10 to 20 Hz with a nearly constant accuracy thereafter for all classifiers. This correlates with the statement in [13] that a sampling frequency above 20 Hz has only a marginally effect on the classification accuracy. For sampling frequencies above 20 Hz only minor improvements can be reported.

4.1.2. Dataset Opportunity

The ANOVA results (presented in Table 3), excludes KStar for the same reason that is mentioned in Section 4.1.1, showed that 53% of the variations in the dependent variable (CA) are described by the four input parameters. As before, other input parameters that were not tested in the scope of this experiment may have further influence. Compared to the table earlier, the Sig. column shows this time, that the three two-way interaction effects (SF and SM, SF and CM, SF and WS) are non-significant for this dataset. The data is also sorted based on the Type III Sum of Squares for an easier overview of the importance of the main and two-way interaction effects. The influential factors of CM, SM, WS, and SF in decreasing order are the same compared to the earlier Table 2 but the order of the two-way interaction effects changed. The significant effect is now SM and WS, followed in decreasing order by SM and CM, and WS and CM with the rest non-significant.

Figure 9 investigates the interaction between SM and WS. It is noticeable that an increased window size decreases the accuracy for each segmentation method. While SWAB shows a near linear decrease in CA, the FOSW and FNSW segmentation methods, show higher variation in CA for longer WS.

Table 3. ANOVA output for the CA as the dependent variable (Tests of Between-Subjects Effects. Dependent Variable: CA).

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	1,785,658 [a]	670	2,665	45	0.000
Intercept	147,805,613	1	147,805,613	2,587,861	0.000
SF	9,501	5	1,900	33	0.000
WS	183,628	31	5,923	104	0.000
SM	639,572	5	127,914	2,240	0.000
CM	759,536	7	108,505	1,900	0.000
SF * SM	778	25	31	0.545	0.968
SF * CM	1,445	35	41	0.723	0.886
SF * WS	4,898	155	32	0.553	1.000
WS * CM	25,038	217	115	2	0.000
SM * CM	70,397	35	2,011	35	0.000
SM * WS	90,865	155	586	10	0.000
Error	1,540,791	26,977	57		
Total	151,132,062	27,648			
Corrected Total	3,326,449	27,647			

[a] R Squared = 0.537 (Adjusted R Squared = 0.525).

Figure 9. Two-way interaction effect for WS and SM.

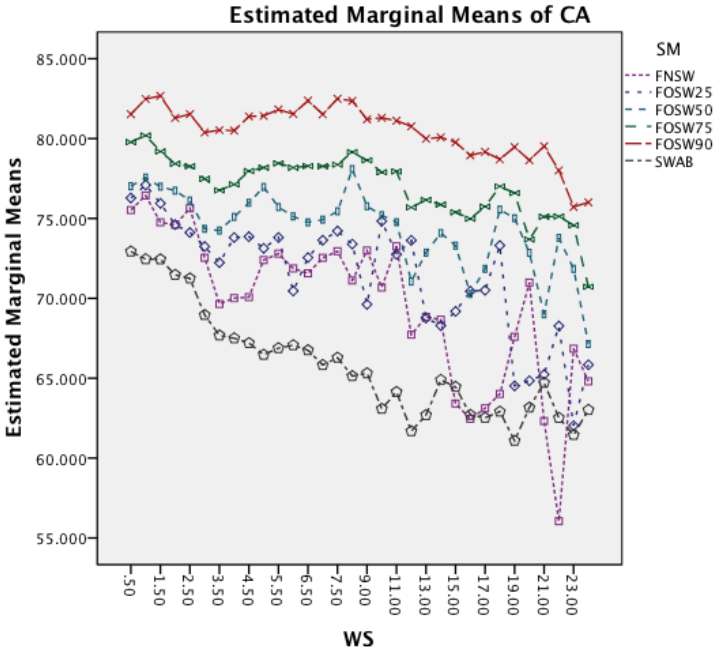
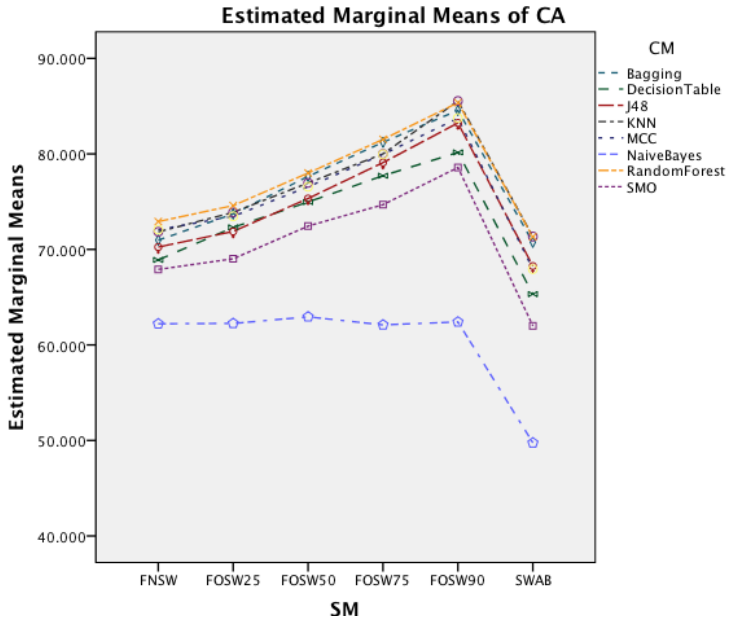


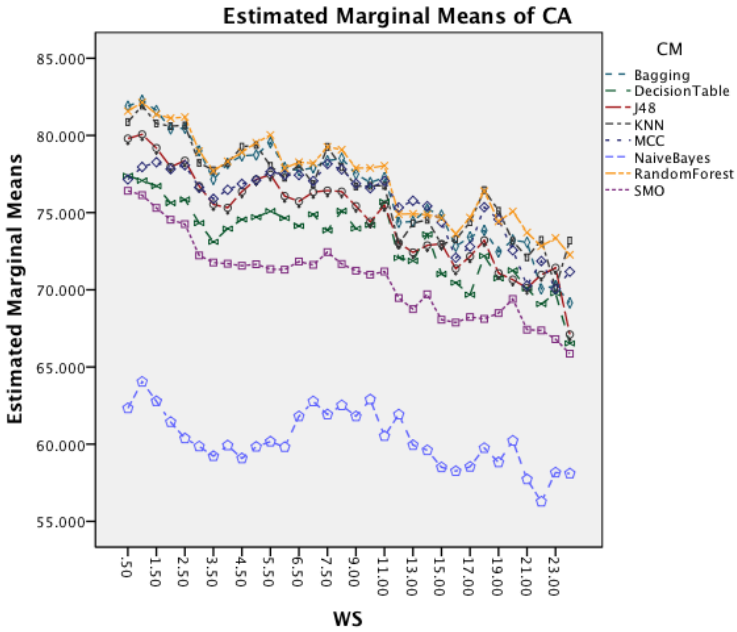
Figure 10 shows the interaction effect of changes in the SM on the CM. The Naïve Bayes classifier shows only a minor improvement in accuracy, whilst the remaining seven classifiers show a substantial increase with an increased segmentation overlap for FOSW. Another visible effect is the poor performance of classifiers with the SWAB segmentation method for this dataset. Results are below FNSW, while earlier it was mostly outperforming FOSW with 75% overlap.

Figure 10. Two-way interaction effect for SM and CM.



The next effect investigated is the interaction between WS and CM in Figure 11. All CM show a decrease in CA for longer WS. The effect is less significant on Naïve Bayes compared to the other classifiers. The graph shows that shorter WS result in better CA result.

Figure 11. Two-way interaction effect for WS and CM.

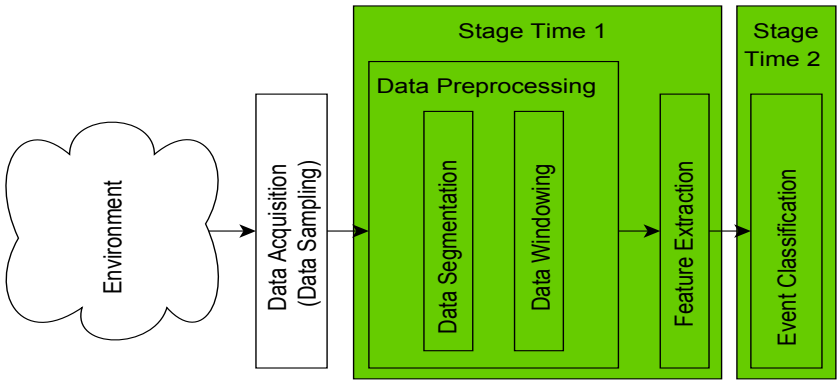


4.2. Statistical Analysis of Computational Load

The selection of different SF, SM, WS and CM does not only have an impact on the classification accuracy but also on the CL of the system. The CL for the classification of ADL events is based on two main factors. The first one is the data pre-processing and feature extraction step (indicated as

$Stage_{Time1}$ in Figure 12) and the second factor is the actual event classification (indicated as $Stage_{Time2}$ in Figure 12).

Figure 12. Timing factor for computational load.



$Stage_{Time1}$ depends on the selected SF, SM and WS, excluding any other pre-processing steps such as filtering which is not of interest in this study, while $Stage_{Time2}$ purely depends on the selected CM. For real-time applications, the combination of SM and WS introduces a limitation for certain parameter combinations, leading to the requirement that Equation (10) needs to be fulfilled:

$$Stage_{Time1} + Stage_{Time2} \leq Max\ CL \quad (10)$$

The authors therefore conducted an analysis with the CL as the dependent variable to investigate the influence of the four input parameter SF, SM, WS and CM. In the preliminary analysis, one of the levels of the SM input showed to have a high influence on the dependent variable. As before with KStar superimposing on parameters for the accuracy, the SWAB segmentation method increases noticeably the CL as compared to the other methods. Hence, effects that were non-significant before are significant once SWAB is removed as a SM level. Therefore, the analysis will outline the overall input variables without SWAB segmentation method.

4.2.1. Dataset Bao *et al.*

The result of the ANOVA is represented in Table 4, outlining that 48% of the variation in the dependent variable CL are described by the variation in the input parameters. The Source column is sorted based on the Sum of Squares to allow for an easier recognition of the importance of an input parameter. The data highlights that the most important factor for the CL is CM (with SWAB included, this effect was actually non-significant). This is followed by WS, SF and as the least significant parameter SM. For the two-way interaction effect the five significant combinations are WS and CM, SM and CM, SF and WS, SM and WS followed by SF and SM.

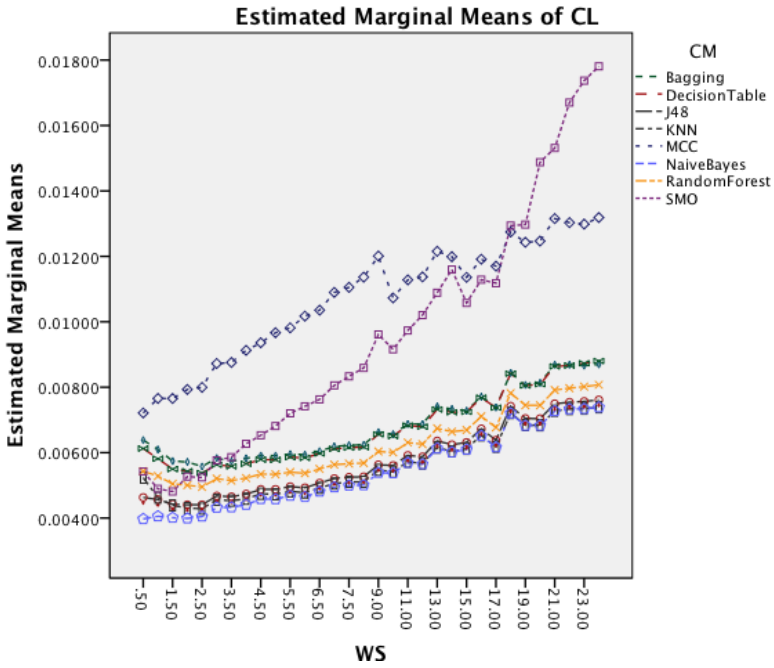
Figure 13 shows the interaction effect between WS and CM. All classifiers require a longer CL for longer WS with a small visible drop in CL for short window sizes about 2 s. The graph also shows that the rate in which the CL increases is higher for MCC and SMO.

Table 4. ANOVA output for the CL as the dependent variable (Tests of Between-Subjects Effects. Dependent Variable: CL).

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	1.316 [a]	626	0.002	222	0.000
Intercept	7.755	1	7.755	818,571	0.000
SM	0.013	4	0.003	355	0.000
SF	0.104	5	0.021	2,202	0.000
WS	0.330	31	0.011	1,122	0.000
CM	0.515	7	0.074	7,769	0.000
SF * CM	0.000	35	3.587×10^{-6}	0.379	1.000
SF * SM	0.001	20	5.456×10^{-5}	5.759	0.000
SM * WS	0.034	124	0.000	29	0.000
SF * WS	0.082	155	0.001	56	0.000
SM * CM	0.103	28	0.004	390	0.000
WS * CM	0.134	217	0.001	65	0.000
Error	1.449	152,957	9.474×10^{-6}		
Total	10.521	153,584			
Corrected Total	2.766	153,583			

[a] R Squared = 0.476 (Adjusted R Squared = 0.474).

Figure 13. Two-way interaction effect for WS and CM.



The interaction effect for segmentation method and classifier method in Figure 14 shows that there is significant improvement in CL for MCC and SMO for an increased overlap. The remaining classifiers show only minor changes.

Figure 14. Two-way interaction effect for SM and CM.

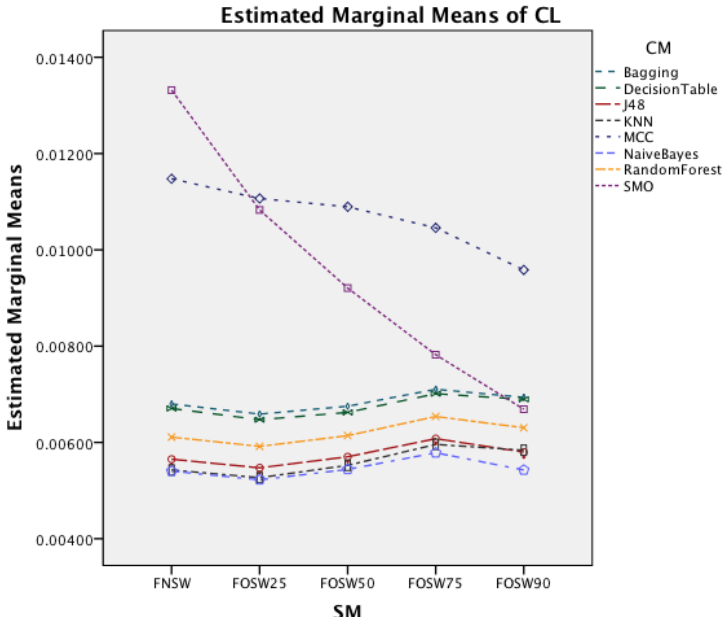
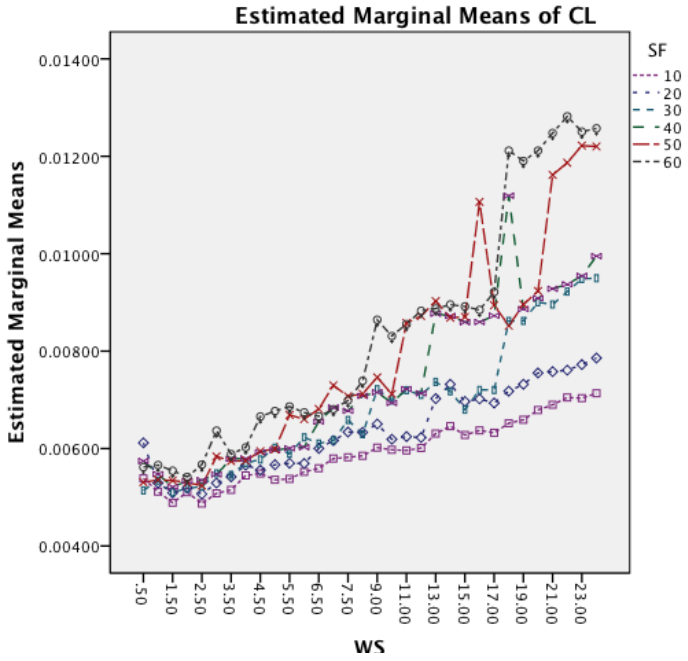


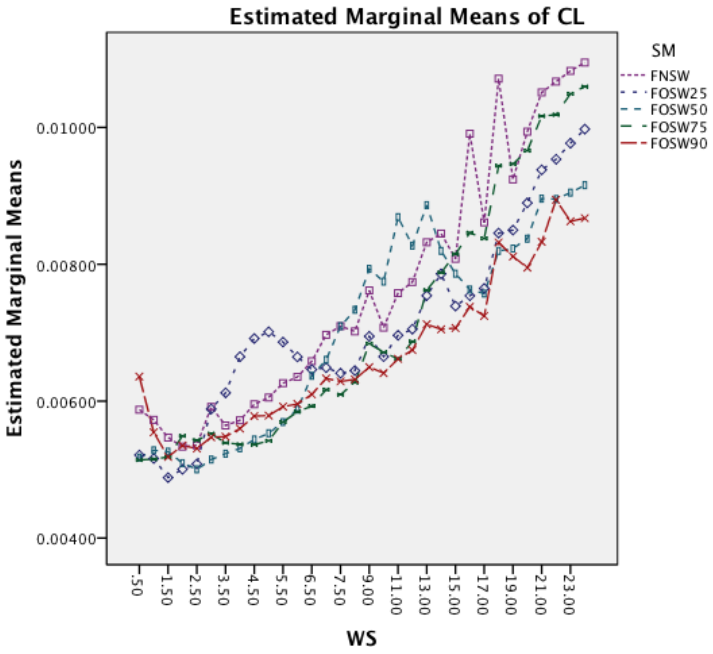
Figure 15 shows the interaction effects between WS and SF. The longer window sizes result in a higher CL in all SF. Moreover, the graph shows that for higher SF the rate of increase in CL does also increase.

Figure 15. Two-way interaction effect for WS and SF.



The last interaction effect under investigation is WS and SM. The graph in Figure 16 shows that the segmentation method has similar patterns to the classifier. All segmentation methods have a significant increase in CL for longer window sizes. An interesting observation is that segmentation methods with higher overlap result in lower CL for higher window sizes.

Figure 16. Two-way interaction effect for WS and SM.



4.2.2. Dataset Opportunity

The result of the ANOVA is represented in Table 5, outlining that 72% of the variation in the dependent variable CL are described by the variation in the input parameters.

Table 5. ANOVA output for the CL as the dependent variable (Tests of Between-Subjects Effects. Dependent Variable: CL).

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	0.358 [a]	626	0.001	93	0.000
Intercept	1.525	1	1.525	247,087	0.000
SM	0.008	4	0.002	328	0.000
SF	0.021	5	0.004	671	0.000
WS	0.090	31	0.003	470	0.000
CM	0.112	7	0.016	2,595	0.000
SF * SM	0.000	20	7.884×10^{-6}	1	0.181
SF * CM	0.000	35	7.236×10^{-6}	1	0.223
SM * WS	0.006	124	4.866×10^{-5}	8	0.000
SF * WS	0.011	155	6.966×10^{-5}	11	0.000
SM * CM	0.050	28	0.002	290	0.000
WS * CM	0.060	217	0.000	45	0.000
Error	0.138	22,413	6.172×10^{-6}		
Total	2.021	23,040			
Corrected Total	0.496	23,039			

[a] R Squared = 0.721 (Adjusted R Squared = 0.714).

The Source column is sorted based on the Sum of Squares to allow for an easier recognition of the importance of an input parameter. The data highlights that the most important factor for the CL is CM. This is followed by WS, SF and as the least significant parameter SM. For the two-way interaction effect the four significant combinations are WS and CM, SM and CM, SF and WS, followed by SM and WS. The interaction effect SF and SM that was significant earlier, is non-significant for this dataset.

Figure 17 shows the interaction effect between window size and classifiers. All classifiers require a longer CL for longer WS. The graph also shows that the rate in which the CL increases is higher for SMO. MCC that had an increased rate in the earlier dataset follows now the behavior of the other classifiers.

Figure 17. Two-way interaction effect for WS and CM.

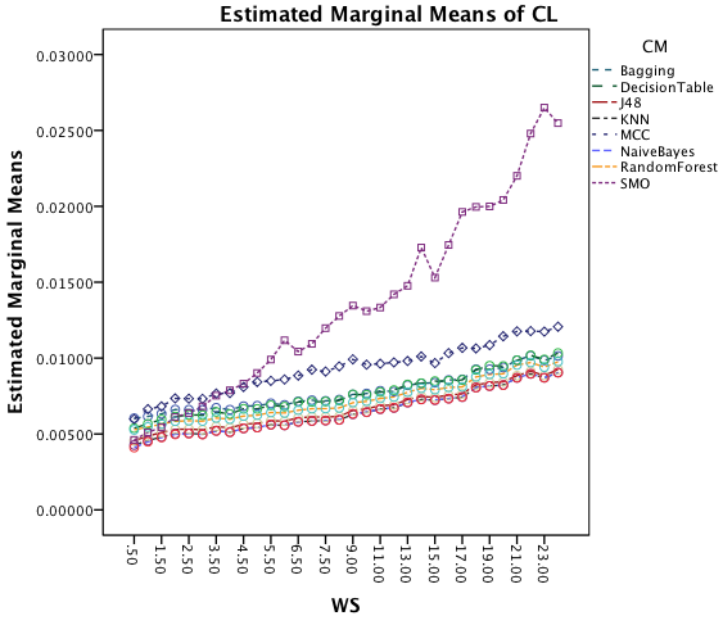
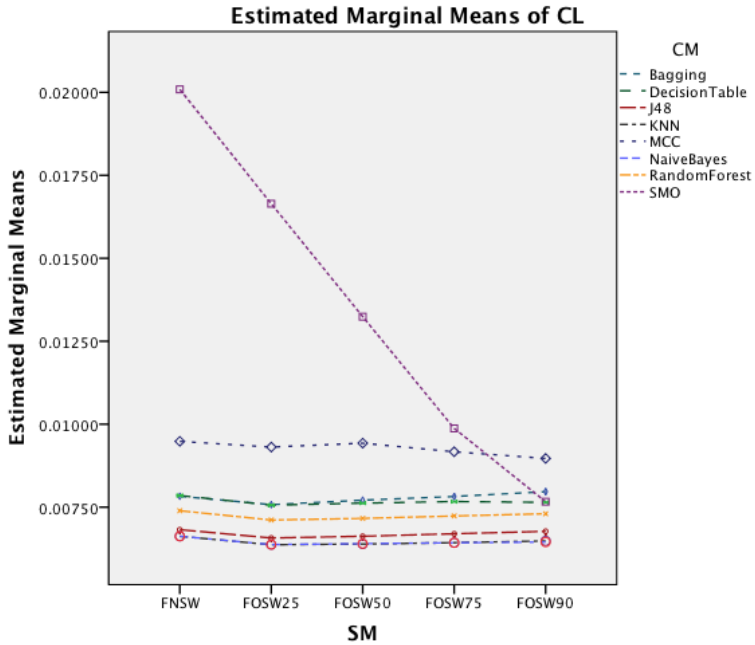


Figure 18. Two-way interaction effect for SM and CM.



The interaction effect for segmentation method and classifier in Figure 18 shows that there is significant improvement in CL for SMO for an increased overlap. The remaining classifiers show only minor changes for the different segmentation methods.

Figure 19 shows the interaction effects between WS and SF. The longer window sizes result in a higher CL for all SF. Moreover, the graph shows that higher sampling frequencies result in an increased rate of CL as well.

Figure 19. Two-way interaction effect for WS and SF.

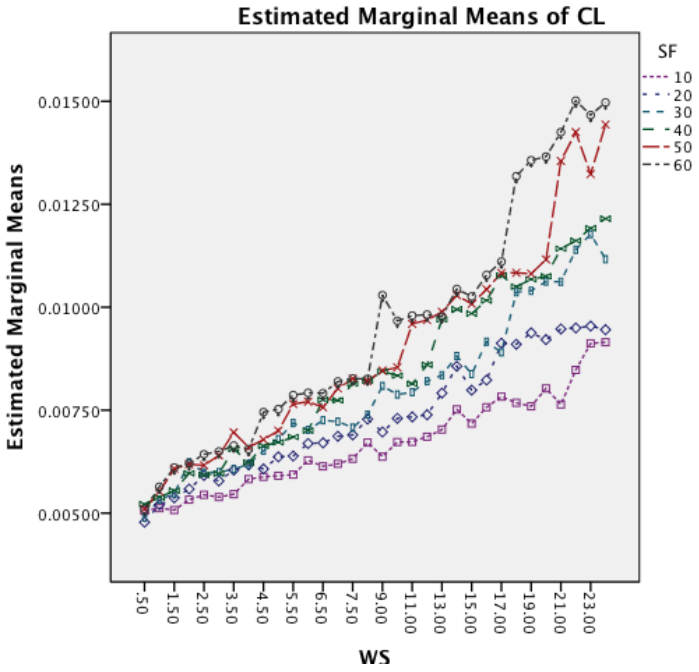
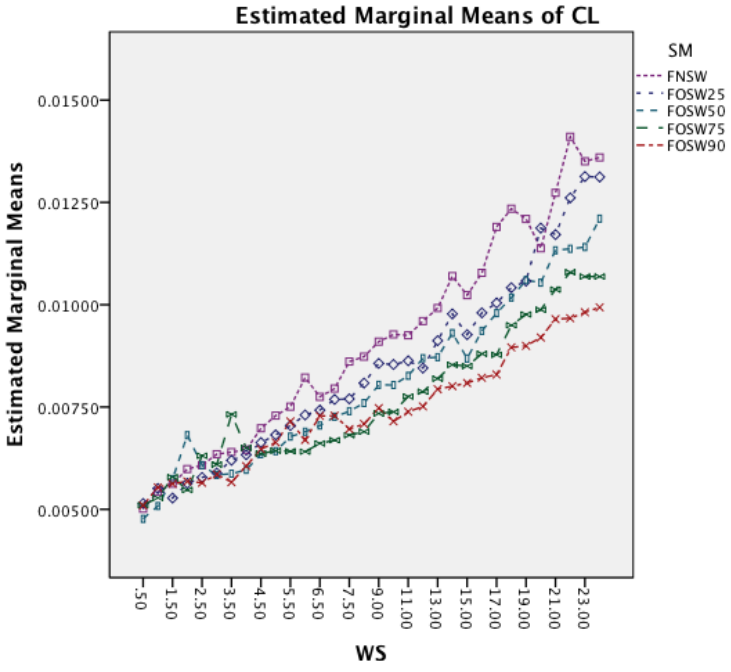


Figure 20. Two-way interaction effect for WS and SM.

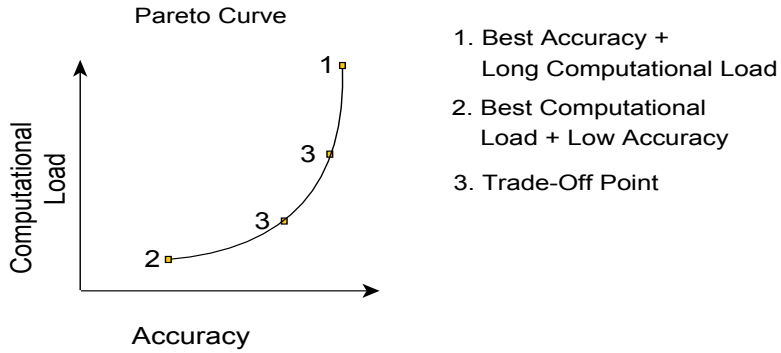


The last interaction effect under investigation is windows size and segmentation method. The graph in Figure 20 shows that the segmentation method has similar patterns to the classifier. All segmentation methods have a significant increase in CL for longer window sizes. An interesting observation, as with the other dataset, is that the segmentation methods with higher overlap result in lower CL for higher window sizes.

4.3. Parameter Selection

Based on the parameter influence described in Sections 4.1 and 4.2, the inevitable question still stands: what is the best parameter selection for a given requirement? The answer, however, depends strongly on the preference with respect to classification performance, e.g., is the best accuracy required or are there limitations to CL. Therefore, a set of well-performing parameter sets based on the trade-off between accuracy and CL were identified. For the given dataset certain parameter combinations will achieve a similar CA but will require different CL and vice versa. When plotted in a graph, such as presented in Figure 21, the best accuracy for a given CL would follow the black line (called Pareto frontier), with dominated parameter sets lying on the left hand side of the curve. Hence, a parameter set is dominated if there exists a combination of parameter values that results in the same level of accuracy with less CL or achieve better accuracy with the same CL. The Pareto frontier, also referred to as Pareto curve, outlines the set of non-dominated solutions, herein represented by a set of parameter combinations. One set of parameter values may achieve best CA at the cost of a high CL (Point 1) and another combination will achieve the lowest CL at the cost of a lower accuracy (Point 2). Parameter sets in between points 1 and 2 on the Pareto frontier are subject to a trade-off (Point 3), hence accepting the sacrifice of either, accuracy or CL, depending on the context of the applications or potential corresponding limitations, e.g., hardware constraints.

Figure 21. Explanation of the Pareto curve.

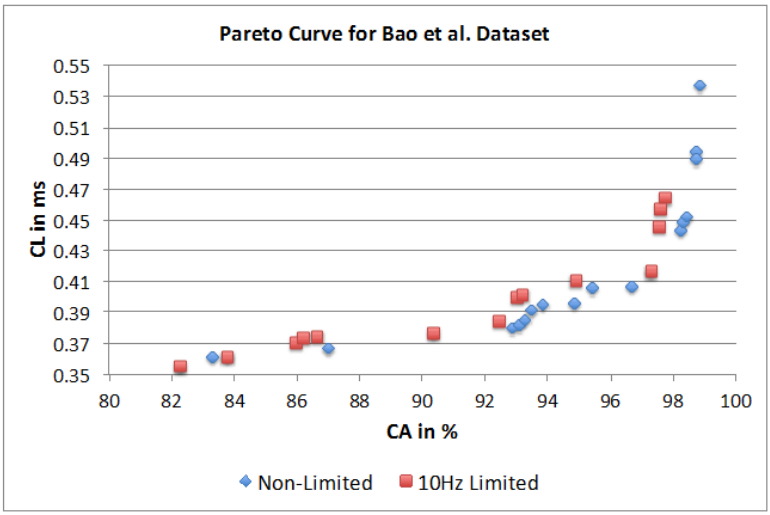


4.3.1. Dataset Bao *et al.*

Figure 22 represents two separate Pareto curves for the Bao *et al.* dataset. The illustration shows a non-limited Pareto curve over all 10,368 possible parameter combinations and a Pareto curve that is limited to 10 Hz sampling frequency, as this is a common hardware limitation for researchers using off-the-shelf components. For the non-limited Pareto curve, with 10,368 possible combinations, only 20 are dominating. The parameter combinations show a maximum SF of 30 Hz with equal occurrence

of 20 and 30 Hz (8 combinations each). For the SM parameter only the FOSW method with various overlaps percentages shows to be of importance. An overlap value of 90% shows higher CA, while an overlap value of 25% results in shorter CL. The dominating points highlight short window sizes of 1.5 s for the shortest CL and a WS of 7.5 s for best CA. The dominating classifiers are Naïve Bayes and KNN. The latter is in the majority of combinations and results in the best CA, while the former results in the shortest CL. The achievable CA ranges from 82.2% with a CL of 0.355 ms to 98.8% with a CL of 0.537 ms. The parameter combinations for the Pareto curve limited to 10 Hz SF, highlights only 14 dominating parameter combinations. The SM shows similar behavior to the non-limited dataset, as well as a higher influence of the FOSW method with 75% overlap. The WS is around 1.5 s for shortest CL and around 7.5 s for best CA. The CM parameter for the 10 Hz limited case is nearly identical to the non-limited case; with the exception of the presence of J48 as an influential algorithm in the former case. The achievable CA ranges from 82.2% with a CL of 0.355 ms to 97.7% with a CL of 0.464 ms.

Figure 22. Dominant points on the Pareto curve.

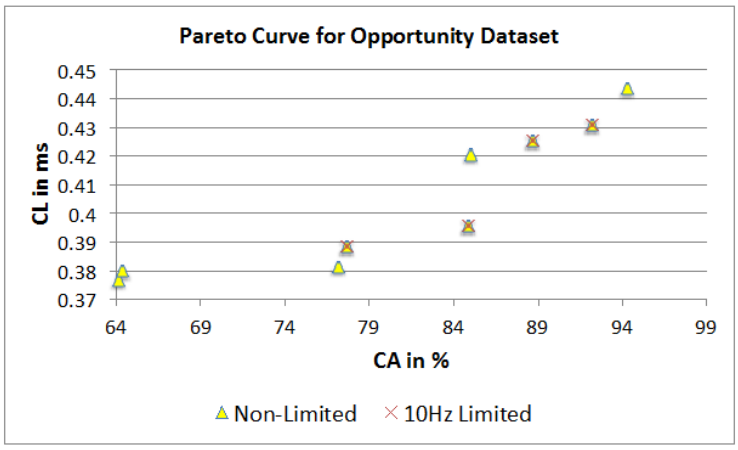


4.3.2. Opportunity Dataset

Figure 23 represents the two separate Pareto curves for the Opportunity dataset, based on a non-limited and 10 Hz limited parameter combination set. For the non-limited Pareto curve, 12 dominating points are identified, while the 10 Hz limited dataset shows only six dominating points. The full dataset shows SF parameter values in the range of 10 to 30 Hz with a majority split between 10 and 20 Hz. FOSW with 90% overlap (SM) dominates for higher CA and with 25% overlap for lower CL in both datasets.

Short WS have a significant influence in this dataset, as both datasets include parameter combinations with 1.5 s or less. The dominating classifiers are Naïve Bayes, and KNN. This is a slight variation compared to the 10 Hz limited Bao *et al.* dataset. The non-limited dataset has a range of 64.2% to 94.3% in CA with a CL of 0.376 to 0.443 ms. The 10 Hz limited dataset ranges from 63.2% to 92.2% for CA with 0.383 to 0.43 ms for CL.

Figure 23. Dominant points on the Pareto curve.



5. Summary and Discussion of Results

One of the main problems in AAL is the availability (or the lack thereof) of test subjects, as compared to clinical trials, where subjects can reach into the thousands. In [41] the authors highlight that research in AAL starts out as a demonstration of feasibility under laboratory conditions, which in a further step needs an increased number of participants and ethical considerations. In [42], the authors argue that the use of any one of two activity classification methods, uniform (where the training data comes from all tests subjects) and individual (training data representing separate test subjects) can lead to problems; generalization (arising from the uniform method), small training data set (individual) can both result in poor performance. The research and associated experiments presented here fall in the individual category as performance measures (CA and CL) are generated for each of the test subjects involved. The authors believe that despite the pitfalls described above, this was the better method to adopt; this is in line with Elbert *et al.*' approach. Moreover, as the tested activities are nearly equally represented in the dataset, using the accuracy measure can be done without loss of validity. This is in contrast with the differentiation between, say, normal and abnormal conditions, where the latter occurs rarely resulting in an imbalanced set of data; the use precision, recall and f-measure would be a more appropriate performance indicator [20].

In summary the outputs of the work presented here, are listed below:

- The importance of parameters for CA ranked in order of decreasing influence is CM, SM, WS and SF;
- The impact of WS is different for both datasets;
- Increased segmentation overlap improves CA;
- The influence of SWAB on CA is different in both datasets;
- SF above 10 Hz has only a minor improvement on CA;
- CL behaves the same for both dataset;
- The importance of parameters for CL ranked in order of decreasing influence is CM, WS, SF and SM;
- Some dominant parameter combinations of the Pareto curve are similar for both datasets;
- Higher CL does not automatically result in higher CA.

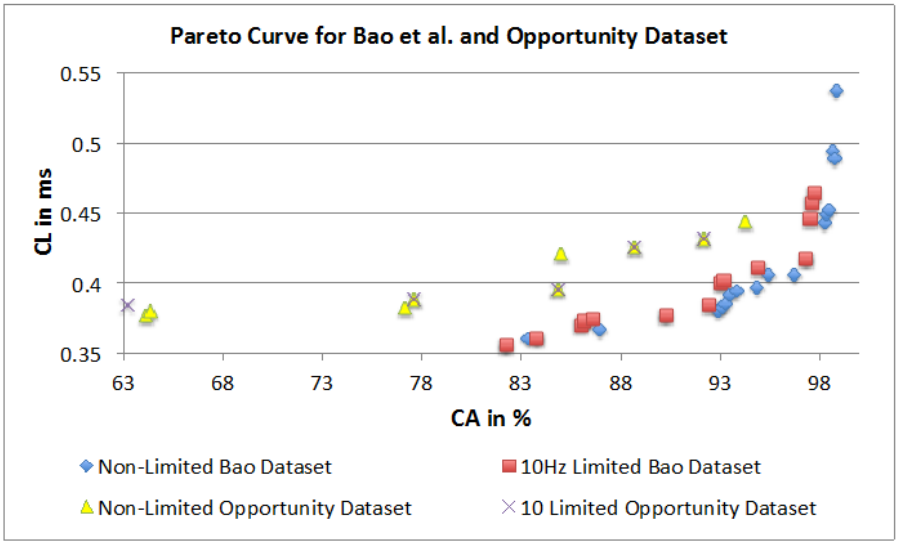
The following discussion will look into the results of the ANalysis Of VAriance (ANOVA) for CA and CL and finish with the dominant parameter points of the Pareto curves. The two-way interaction effect between SM and CM highlights for both datasets that FOSW with 90% overlap results in the best CA. From FNSW (no overlap) to FOSW with 90% overlap, both datasets show that more overlap improves the CA. A possible reason for this is that the increase in overlap allows for a bigger training set and has the lowest loss of information, in the range of investigated SM. The results for SWAB are mixed. For the Bao *et al.* dataset CA is just below CA for FOSW with 90% overlap, while the Opportunity dataset showed SWAB to be the worst segmentation method tested. Further research needs to look into the actual benefit of a dynamically sliding window, which incidentally was reported in [20] as giving good results, as the results reported here (in terms of classification accuracy) are inconsistent between the two datasets. Another difference between the datasets is observed for the WS and CM two-way interaction effect. While for the first dataset (Bao *et al.*), the CA improves for window sizes between 1 and 8 s and only decrease for WS values above 8 s, the second dataset (Opportunity) achieves best CA for 0.5 s and starts to decrease immediately after that. A similar behavior can be seen for the two-way interaction effect of WS and CM and WS and SM with both datasets (compare Figure 6 and Figure 7 in Section 4.1.1 and Figure 11 and Figure 9 in Section 4.1.2). Researchers should therefore choose smaller window sizes if possible. Another difference between the two datasets is the significance of the two-way interaction effect WS and SF; namely, significant interaction (but not the most significant) for Bao *et al.* and non-significant interaction for the Opportunity dataset. The graph in Figure 8 (see Section 4.1.1) shows that sampling frequencies above 10 Hz achieve nearly the same CA, while the 10 Hz sampling frequency is marginally lower, endorsing the finding in [13] that sampling frequencies above 20 Hz result in only minor accuracy gains.

For CL, both datasets show the same behavior for the two-way interaction effects. For the three interaction effects including WS (WS and CM, WS and SM, WS and SF) similar behavior is observable. A shorter WS results in a lower CL, while a longer WS will increase the CL. This effect is lowest for WS and CM and highest for WS and SM. The interaction effect between SM and CM highlights no significant change for any classifier besides SMO. SMO is the only classifier that can reduce the CL with an increased segmentation overlap.

The authors used ANOVA to quantify the influence of the different parameters on the CA and CL. They have also used a Pareto curve based approach to highlight dominant parameter combinations for "optimum" achievable performance (optimality being decided by the user in a given context/application). Figure 24 presents the four Pareto curves based on the dominant combinations. The illustration shows that all graphs have a similar outline and that it is possible to achieve similar results irrespective of the dataset. This is highlighted with only a 4.3% difference in CA between the two top performing parameter combinations. However, the dominant parameter combinations are different for each dataset. Therefore, it is not possible to present a single combination that will work best for all datasets. Having said that, some dominant points have similar parameter combinations. In both datasets high CA is achieved with the KNN classifier and the FOSW with 90% overlap for SM. Furthermore, the Pareto points show that a sampling frequency above 30 Hz is not necessary and only minor improvements in CA are achieved with a sampling frequency above 10 Hz. As a consequence, the authors recommend adjusting parameters individually for each dataset and test subject to achieve optimal results, especially with regards to WS. The Pareto curves also reveal that a higher

computational load does not necessarily result in better classification accuracy, as the algorithms under investigation are not recursive. The Pareto curve is also a good tool to investigate the influence of a hardware limitation such as a low sampling rate, storage space and battery runtime. When superimposing the hardware limited Pareto curve with the non-limited curve a simple comparison of achievable CA and CL is possible. The results presented in Section 4.3 in combination with the ANOVA in Sections 4.1 and 4.2 can be used for future research as a tool to select parameter combinations for AAL event classifications with the sound understanding of how each parameter influences the outcome of event classification accuracy and computational load.

Figure 24. Dominant points on the Pareto curve for both datasets.



6. Conclusions and Future Work

This paper has presented a new instrument to help select data capture and processing parameters for the recognition of Activities of Daily Living (ADL). A review of the literature uncovered a lack of consensus in terms of the selection of sampling frequency, segmentation method and window size, and classifier method for the recognition of ADL. The impact of the sampling frequency (six levels), segmentation method (three segmentation algorithms with different parameters resulting in six different levels) and segmentation window size (32 levels) on the classification accuracy and computational load of a set of commonly used classifiers (nine levels) has been investigated. This has involved experimenting with two datasets, containing 20 and three test subjects, respectively, and analysis of the resulting data using ANalysis Of VAriance (ANOVA). The analysis showed that the choice of classifier method is the most important parameter followed by the segmentation method, window size and finally sampling frequency. It also showed that in the case of computational load the parameters ranked in order of decreasing influence are classifier method, window size, sampling frequency, and segmentation method. The results have been presented graphically using a Pareto curve, which highlighted two dominant classifiers for both datasets (KNN, Naïve Bayes). The Pareto curve did not show matching dominant points in both datasets, however, it showed that combinations of three out of the four factors (CM, SM, SF) are likely to result in dominant points. The authors have

suggested that the Pareto curve is a good instrument which can be used to select sets of parameters based on their impact on classification accuracy and computational load and resolve trade-off issues.

As part of their future work in the general area of AAL, the authors plan to investigate a number of issues specific to the findings presented in this paper. An important point of interest is the identification of the reasons behind the inconsistency between the two datasets used in terms of the influence of WS on the classification accuracy. A possible influential factor, not considered in the present work, is the nature of the ADL itself. It might be necessary to adjust the WS parameter with regards to the expected ADLs in the dataset; [16] suggested to use different WS parameter combinations per activity. The authors also intend to investigate the influence of the features extracted and the position of sensors on classification accuracy. Different feature combinations (and a reduction in the number of required features) may improve the classification accuracy of different ADLs as well as reduce the CL. Moreover, the authors' propose to couple the results obtained so far with a Decision Support System (DSS). Having the option to learn, adjust from past experiences, and include new ADLs, would allow for more informed decisions in parameter selection over time. Additionally, hardware limitations, such as battery time and communication bandwidth, should be included into the selection process. Another direction that the authors want to pursue is the investigation of how to improve the Pareto curve by replacing the computational load with a measure for training time and training samples, as it could highlight classifiers that could achieve good accuracy within a low starting time.

Author Contributions

Sebastian D. Bersch, Ifeyinwa E. Achumba and Djamel Azzi designed the experiment, Sebastian D. Bersch performed the experiments, Sebastian D. Bersch, Jana Ries, Djamel Azzi and Rinat Khusainov analyzed the data, Sebastian D. Bersch, Jana Ries, Ifeyinwa E. Achumba, Djamel Azzi and Rinat Khusainov prepared the manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Amft, O.; Tröster, G. Automatic dietary monitoring: On-body sensing domains. In *Automatic Dietary Monitoring Using On-Body Sensors*; Ph.D. Thesis, ETH Zürich, Zürich, Switzerland, 2008.
2. Sixsmith, A.J. An evaluation of an intelligent home monitoring system. *J. Telemed. Telecare* 2000, *6*, 63–72.
3. Pietka, E. Expert systems in parameter extraction of the ECG signal. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, New Orleans, LA, USA, 4–7 November 1988; pp. 165–166.
4. Orfanidis, S.J. *Introduction to Signal Processing*; Prentice-Hall: Upper Saddle River, NJ, USA, 1995.

5. Huynh, T.; Schiele, B. Analyzing Features for Activity Recognition. In Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies, Grenoble, France, 12–14 October 2005; pp. 159–163.
6. Sekine, M.; Tamura, T.; Fujimoto, T.; Fukui, Y. Classification of Walking Pattern Using Acceleration Waveform in Elderly People. In Proceedings of the 22nd Annual IEEE International Conference of the Engineering in Medicine and Biology Society, Chicago, MI, USA, 23–28 July 2000; Volume 2, pp. 1356–1359.
7. Bao, L.; Intille, S. Activity recognition from user-annotated acceleration data. In *Pervasive Computing*; Springer: Berlin, Germany, 2004; pp. 1–17.
8. Preece, S.J.; Goulermas, J.Y.; Kenney, L.P.; Howard, D. A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Trans. Biomed. Eng.* 2009, *56*, 871–879.
9. Wang, N.; Ambikairajah, E.; Lovell, N.H.; Celler, B.G. Accelerometry Based Classification of Walking Patterns Using Time-Frequency Analysis. In Proceedings of 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Lyon, France, 22–26 August 2007; pp. 4899–4902.
10. Casale, P.; Pujol, O.; Radeva, P. Human activity recognition from accelerometer data using a wearable device. *Pattern Recognit. Image Anal.* 2011, *6669*, 289–296.
11. Ravi, N.; Dandekar, N.; Mysore, P.; Littman, M. Activity Recognition from Accelerometer Data. In Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence, Menlo Park, CA, USA, 9–13 July 2005; pp. 1541–1546.
12. Parkka, J.; Cluitmans, L.; Ermes, M. Personalization algorithm for real-time activity recognition using PDA, wireless motion bands, and binary decision tree. *IEEE Trans. Inf. Technol. Biomed.* 2010, *14*, 1211–1215.
13. Maurer, U.; Smailagic, A.; Siewiorek, D.P.; Deisher, M. Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions. In Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks 2006, Boston, MA, USA, 3–5 April 2006; pp. 113–116.
14. Antonsson, E.K.; Mann, R.W. The frequency content of gait. *J. Biomech.* 1985, *18*, 39–47.
15. Bouten, C.V.; Koekkoek, K.T.; Verduin, M.; Kodde, R.; Janssen, J.D. A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *IEEE Trans. Biomed. Eng.* 1997, *44*, 136–147.
16. Gjoreski, H.; Gams, M.; Chorbev, I. 3-axial accelerometers activity recognition. *ICT Innov. (2010)* 2010, 51–58.
17. Keogh, E.; Chu, S.; Hart, D.; Pazzani, M. An Online Algorithm for Segmenting Time Series. In Proceedings of the International Conference on Data Mining, San Jose, CA, USA, 29 November–2 December 2001; pp. 289–296.
18. Chu, C.-S.J. Time series segmentation: A sliding window approach. *Inf. Sci.* 1995, *85*, 147–173.
19. Kozina, S.; Lustrek, M.; Gams, M. Dynamic signal segmentation for activity recognition. In Proceedings of International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011; pp. 15–22.

20. Ortiz, J.; Olaya, A.G.; Borrajo, D. A Dynamic Sliding Window Approach for Activity Recognition. In Proceedings of the International Conference on User Modeling, Adaptation and Personalization, Girona, Spain, 11–15 July 2011; pp. 219–230.
21. Nyan, M.N.; Tay, F.; Seah, K.; Sitoh, Y.Y. Classification of gait patterns in the time-frequency domain. *J. Biomech.* 2006, *39*, 2647–2656.
22. Van Kasteren, T.L.M.; Noulas, A.; Englebienne, G.; Kröse, B.J. Accurate Activity Recognition in a Home Setting. In Proceedings of the Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Seoul, Korea, 21–24 September 2007; pp. 1–9.
23. Patterson, D.J.; Fox, D.; Kautz, H.; Philipose, M. Fine-Grained Activity Recognition by Aggregating Abstract Object Usage. In Proceedings of the International Semantic Web Conference (ISWC), Galway, Ireland, 18–21 October 2005; pp. 44–51.
24. Roggen, D.; Calatroni, A.; Rossi, M.; Holleczek, T.; Forster, K.; Tröster, G.; Lukowicz, P.; Bannach, D.; Pirkl, G.; Ferscha, A.; *et al.* Collecting complex activity datasets in highly rich networked sensor environments. In Proceedings of 2010 7th International Conference on Networked Sensing Systems (INSS), Kassel, Germany, 15–18 June 2010; pp. 233–240.
25. MATLAB—The Language of Technical Computing. Available online: http://www.mathworks.cn/products/matlab/ (accessed on 3 March 2014).
26. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.* 2009, *11*, 10–18.
27. Achumba, I.E.; Bersch, S.; Khusainov, R.; Azzi, D.; Kamalu, U. On time series sensor data segmentation for fall and activity classification. In Proceedings of 2012 IEEE 14th International Conference on e-Health Networking, Applications and Services (Healthcom), Beijing, China, 10–13 October 2012; pp. 427–430.
28. Farringdon, J.; Moore, A.J.; Tilbury, N.; Church, J.; Biemond, P.D. Wearable Sensor Badge and Sensor Jacket for Context Awareness. In Proceedings of the IEEE International Symposium on Wearable Computers (ISWC'99), San Francisco, CA, USA, 18–19 October 1999; pp. 107–114.
29. Kawahara, H.S.Y.; Hisashi Kurasawa, H.M.; Aoyama, T. A Context-Aware Collaborative Filtering Algorithm for Real World Oriented Content Delivery Service. In Proceedings of 7th International Conference Ubiquitous Computing, Tokyo, Japan, 11–14 September 2005.
30. Veltink, P.; Bussmann, H.; de Vries, W.; Martens, W.; van Lummel, R. Detection of static and dynamic activities using uniaxial accelerometers. *IEEE Trans. Rehabilit. Eng.* 1996, *4*, 375–385.
31. Jin, G.H.; Lee, S.B.; Lee, T.S. Context awareness of human motion states using accelerometer. *J. Med. Syst.* 2007, *32*, 93–100.
32. Kern, N.; Schiele, B.; Schmidt, A. Recognizing context for annotating a live life recording. *Pers. Ubiquitous Comput.* 2006, *11*, 251–263.
33. Karantonis, D.M.; Narayanan, M.R.; Mathie, M.; Lovell, N.H.; Celler, B.G. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Trans. Inf. Technol. Biomed.* 2006, *10*, 156–167.
34. Bianchi, F.; Redmond, S.; Narayanan, M.; Cerutti, S.; Celler, B.; Lovell, N. Falls event detection using triaxial accelerometry and barometric pressure measurement. In Proceedings of 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Minneapolis, MN, USA, 3–6 September 2009; pp. 6111–6114.

35. Nham, B.; Siangliulue, K.; Yeung, S. *Predicting Mode of Transport from iPhone Accelerometer Data*; CS 229; Machine Learning Final Projects, Stanford University: Stanford, CA, USA, 2008.
36. Entropy (Wavelet Packet). Available online: http://www.mathworks.cn/cn/help/wavelet/ref/wentropy.html (accessed on 3 March 2014).
37. Fast Fourier Transform. Available online: http://www.mathworks.cn/cn/help/matlab/ref/fft.html (accessed on 3 March 2014).
38. Chernbumroong, S.; Atkins, A.S.; Yu, H. Activity Classification Using a Single Wrist-Worn Accelerometer. In Proceedings of 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA), Benevento, Italy, 8–11 September 2011; pp. 1–6.
39. SPSS Software—Predictive Analytics Software and Solutions. Available online: http://www-01.ibm.com/software/uk/analytics/spss/ (accessed on 3 March 2014).
40. Chawla, N.V. Data Mining for Imbalanced Datasets: An Overview. In *Data Mining and Knowledge Discovery Handbook*; Springer US: Boston, MA, USA, 2010; pp. 875–886.
41. Ding, D.; Cooper, R.A.; Pasquina, P.F.; Fici-Pasquina, L. Sensor technology for smart homes. *Maturitas* 2011, *69*, 131–136.
42. Elbert, D.; Storf, H.; Eisenbarth, M.; Ünalan, Ö.; Schmitt, M. An approach for detecting deviations in daily routine for long-term behavior analysis. In Proceedings of 2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), Dublin, Ireland, 23–26 May 2011; pp. 426–433.

Artificial Immune Systems for Anomaly Detection in Ambient Assisted Living Applications

Sebastian Bersch, School of Engineering, University of Portsmouth, Portsmouth, UK

Djamel Azzi, School of Engineering, University of Portsmouth, Portsmouth, UK

Rinat Khusainov, School of Engineering, University of Portsmouth, Portsmouth, UK

Ifeyinwa E. Achumba, School of Engineering, University of Portsmouth, Portsmouth, UK

ABSTRACT

This paper makes a case for the use of Artificial Immune Systems (AIS) in the area of Ambient Assisted Living (AAL) for anomaly detection and long term monitoring. A brief literature review of some of the solutions developed for AAL and the use of AIS in other fields of research is presented. The authors advocate the use of AIS in AAL based on their unique features and their ability to address problems specific to the long term monitoring of people. An improved method for the optimisation of detector generation for AIS, which uses a novel intelligent seeding technique, is presented. The new seeding technique is compared with two other detector seeding methods. The simulation results are presented showing an improvement in the classification accuracy and warranting current and future work.

Keywords: *Ambient Assisted Living (AAL), Anomaly Detection, Artificial Immune System (AIS), Detector Seeding, Long Term Monitoring*

INTRODUCTION

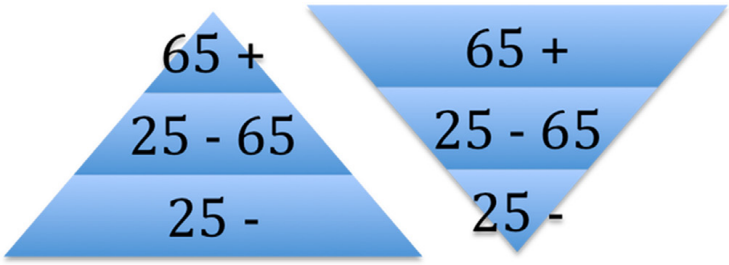
Advances over the last decades in medicine research are helping humans to live longer than ever before, simultaneously; younger people are increasingly reluctant to start a family early, which has resulted in a reduced birth rate. This scenario can be observed in most western countries. Both trends lead to a shift in the current age pyramid (illustrated in Figure 1), which indicate that soon there may be more people over the age of 65 than under(Commission, Economic, & Affairs, 2009).

One consequence of this is that fewer carers have to look after the growing number of elderly and therefore need to be efficiently organized and supported. The research of Ambient Assisted Living (AAL) is centred on finding novel, efficient and cost effective ways to support carers and elderly to stay longer healthy at home.

The work in the area ranges from sensor development for better vital sign monitoring (Boylan, 2011) and position information to recognition of Activities of the Daily Living (ADL) (Sim, et al., 2010) over to health monitor-

DOI: 10.4018/ijaci.2013070101

Figure 1. Age pyramid [Adapted from (Commission, Economic, & Affairs, 2009)]



ing (Monekosso & Remagnino, 2010; Bersch, Chislett, Azzi, & Khusainov, 2011).

Research has tended to concentrate on finding solutions for health and lifestyle problems using short term monitoring. New research on the other hand, is focussed more on the use of long term monitoring of a person (Elbert, Storf, & Eisenbarth, 2011). The authors believe that the ultimate solution will be an adaptive approach that will harness the strength of both short and long term monitoring without the need for user involvement. Therefore understanding what normal behaviour means is a crucial part of the long term monitoring process. Thus the simplest starting point for a long term monitoring is a person without any abnormal behaviour pattern (healthy person) to train the system over a long period of time. The user should only become aware of such a system in a case of an emergency situation later on in their life. The authors propose the use of an artificial intelligence (AI) technique already used in electronic hardware fault and network intrusion detection based on Artificial Immune Systems (AIS) for the purpose of anomaly detection in health monitoring(Bersch, Azzi, & Khusainov, Fall Detection using Biologically Inspired Monitoring, 2011).

RELATED WORK

AIS is an AI method designed to mimic typical functions of the human immune system (HIS) while being oriented at the better known genetic algorithms (GA). The main task of the immune system is to classify body cells in a self set (normal) and a non-self set (abnormal). The framework for this unsupervised classification technique is that normal cells have to be a frequent pattern in the body while abnormal cells have to be the exception in the body. The human immune system uses the chemical surface of a body cell to determine if the cell has been discovered before or not, therefore acting as an anomaly detection system. Different researchers have shown that the characteristics of the human immune system can be modelled using AI terminologies to work outside the human body.

The use of AIS outside the area of AAL is quite varied; some examples are intrusion detection for computer networks, fault detection of electronic machines and unsupervised classification of different data sets. The main connection of these fields are that the AIS is used to "find" the normal self set of the computer network, machine or dataset and trigger an alarm in the case of a deviation from the norm.

In Hofmeyr and Forrest (2000) the authors designed a general purpose architecture for AIS and proofed their concept using a network intrusion detection system (LISYS). This architecture was used as the basic framework for the research described.

Another research team (Zhi-tang, Yao, & Li, 2005) designed a fuzzy anomaly detection algorithm based on AIS to use it for network anomaly detection. Their results showed that the negative selection algorithm can be used to detect anomalies on real network traffic data and the efficiency of the developed algorithm.

Additionally, Golovko et al. proposed to use AIS in another intrusion detection system to detect unknown network attacks in Golovko, Komar, and Sachenko (2010).

In another study Cai et al. (2010) compared Artificial Immune Recognition Systems (AIRS) with Artificial Neural Networks (ANN), Logistic Regression (LR), Support Vector Machines (SVM) and K-Nearest Neighbour (KNN) for their usability in fault detection. The authors used the different machine-learning algorithm to classify real power distribution fault data from three regions in North Carolina. The results showed that AIS based algorithm outperformed the other tested algorithms most of the time.

In the field of domestic heating monitoring Lehmann et al. used AIS to control the heating system of an intelligent home (Bersini & Carneiro, 2006). The aim was to learn the inhabitants' behaviour and adjust the heating pattern for the house accordingly. The AIS was used to act quickly on the users' special heating demands without forgetting its normal behaviour. The author tested the system with the scenarios of regular use, the regular use with a break of two weeks of no use, the regular use with a change of use after a number of weeks and irregular use with frequent changes. The AIS was able to adapt and could still react to changes in the users normal behaviour.

In medical research, AIS scored especially well in the work of (Polat & Gunes, 2006). The authors classified EEG and lung cancer data sets using AIS and were able to achieve 100% accuracy on the data set and pointed out that this accuracy is the highest among the classifier reports in the literature. One reason for these good results is the self-learning ability of AIS since it allows for a fully automated classification.

Gianini et al. proposed to use the AIS paradigm in the area of ambient intelligence, specifically for facial expression detection with additional context information (Gianini, et al., 2009). The authors also investigated human behaviour recognition but did not make any connection to health related monitoring.

Whilst the literature outside the area of AAL showed an effective use of the anomaly detection capabilities of AIS, to the authors' best knowledge the use of AIS in the area of AAL is a completely new approach for the purpose of long term monitoring.

The rest of the paper is organised as follows. The next section highlights the motivation and the design of an AIS for AAL. This is followed by the presentation of the simulation conditions and experimental results testing the performance of the method designed. Finally, the discussion and conclusion are given in the last section.

USING ARTIFICIAL IMMUNE SYSTEM FOR AAL

Motivation

The biggest difference between data collection for long term and short term monitoring in AAL is that the latter is able to produce a much better labelled dataset. This leads to more detailed learning sets for the AI methods used and therefore in better classification results. Collecting such detailed data sets for long term monitoring for a longer period of time is often a much more complex task. The main issues that need to be addressed are:

- Improved hardware design for the longer runtime.
- Knowledge of which data will be meaningful to avoid collection of irrelevant data and enable the optimisation of data processing and storage.
- Time for data collection and manual labelling of the collected data.
- User acceptance of monitoring hardware for long periods of time.

All these points make the long term monitoring a much harder research field; hence the interest of unsupervised learning methods of the authors. Furthermore, all possible health

related changes of an elderly person are hard to foresee, making concrete predictions much more challenging. Additionally, health problems of elderly people do not usually suddenly appear but tend to develop slowly over time. Even falls, which could class as one off conditions, are probably the manifestation of long term conditions which worsen over time. Therefore a change, for a longer period of time, in the person's normal behaviour increases the probability of a future/near health problem (see Figure 2) (Virone, 2008). Once a possible problem is identified, either a carer or a better-suited software agent can do a more detailed diagnostic of the underlying reason for the deviation. The research presented in this paper concentrates on enabling the detection of deviations from normal behaviour and not on the identification of its cause.

Design of AIS

The general structure of an Artificial Immune System is presented in Figure 3. The first task is to define the search space and then to spawn detectors in the defined area, which is equivalent

Figure 2. Health changes over time

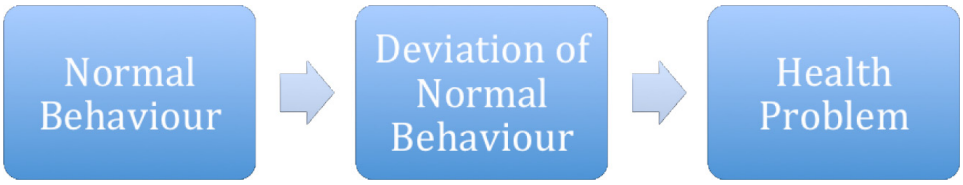
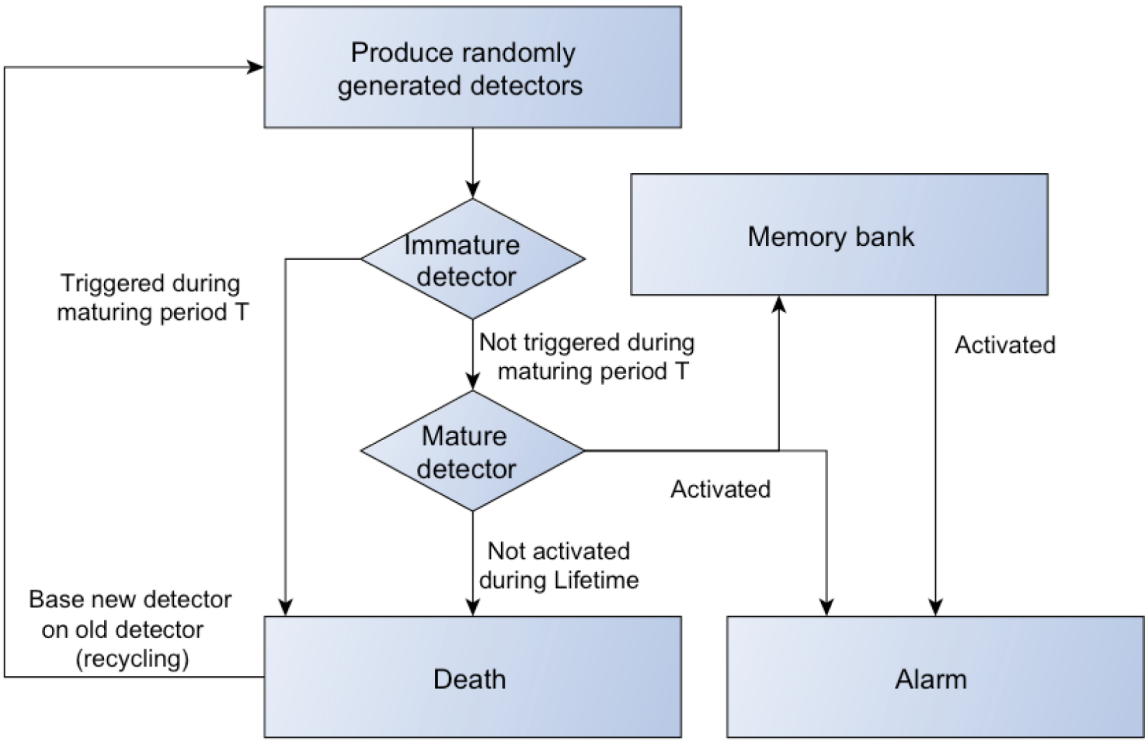


Figure 3. AIS flow diagram [Adapted from (Hofmeyr & Forrest, 2000)]



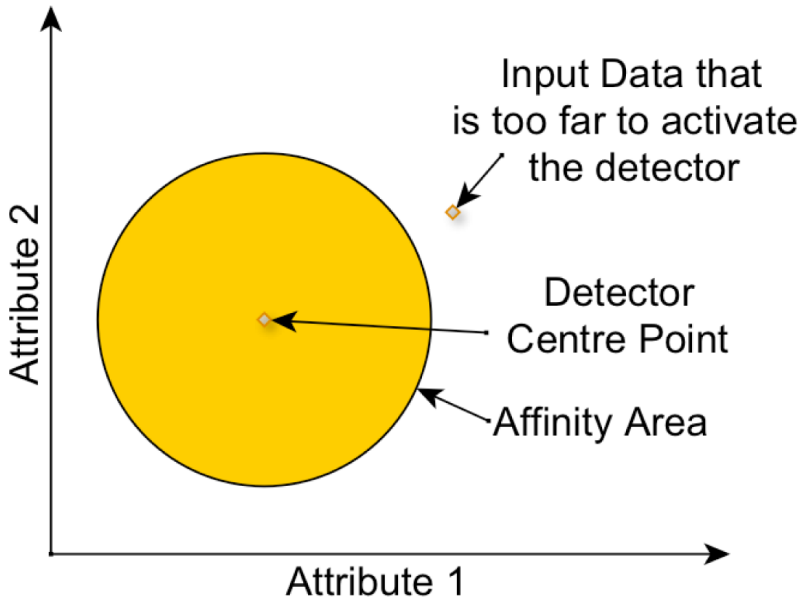
to the production of white blood cells in the human immune system. Such a detector can be in one of three stages:

- Immature
- Mature
- Mature detector stored in the memory bank

Each detector will have a certain affinity area around its centre point, which indicates the detector's activation area. If a data input falls into this radius the detector will be activated (see Figure 4). The authors have used the Euclidean distance to determine the distance from a data input point to a detector centre point.

The assigned stage of a detector, right after the spawning, is the immature level. In this stage the detector cannot raise an alarm in the monitoring system, it is the learning phase for the detector. This stage determines if a detector falls into the "normal" self-set of the data, this is the case if the detector is triggered before it reaches maturity. Should this be the case the easiest solution is to delete the immature detector and replace it with a new one. Another option can be to first reduce the affinity area around the detector (taking into account that it is close to the area of interest) until a threshold is reached specifying that it (the detector) should be removed from the detector pool as well. Detectors that survive this process for a period of time T evolve (mature) and progress to the next stage. Therefore mature detectors are non-activated immature detectors (resulting from a negative selection because the detectors with no affinity to the normal data survives). A mature detector will not have the option to be changed during its lifespan. As a result of this training method, activations thereafter indicate unknown data points or unseen behaviour patterns. At this point, this does not necessarily mean a problem with the monitored person. One reason could be an outlier in the data stream. Should more data points activate mature detectors (fall into their affinity area), the probability that some abnormal activity is taking place increases. Again this does not necessarily indicate a health problem, but might be a change in the routine of the monitored person. The authors expect that the adaptability of the AIS will help to cope with these situations and as mentioned above, the AIS system will be an anomaly detection system for either a carer or for a more specialised agent that has to determine the cause of the change in data.

Figure 4. Explanation of a detectors affinity area



Should a mature detector not be activated during its lifespan, it will be removed from the pool to allow for a new detector to be spawned. This is the only option for detectors to change once they passed the maturing phase. Each destroyed detector will be replaced by a modified immature detector based on the old detector (detector mutation), therefore allowing the AIS to dynamically adjust and adapt to changes in the input data over time without a dedicated relearning phase. This recycling process is not part of the work described here.

The first response to activation is to raise an alarm and then move the activated detector into a memory bank (its final stage). A detector that reaches the last stage should normally stay forever in the memory bank. Reasons against a persistent storage are memory limitations and impact on the adaption feature of AIS to changes in the input stream. Possible options for an improved memory management of the available resources are to let detectors compete for a memory place based on their distance to the abnormal data point or activation frequency and remove memory detectors that have a low activation level or are activated infrequently (Hofmeyr & Forrest, 2000). Furthermore, the affinity area should be further reduced to indicate a specialisation of the detector. The recycling and storage of detectors in a memory bank will not be discussed here.

Aim of this Research

The literature review did not reveal a simple solution for the design of AIS with a good detector generation method to fully cover the search space. It was therefore the aim of the work presented here to investigate the means to achieve good coverage of the search space with a reduced overlap of detectors while maintaining AIS adaptability.

SIMULATION CONDITIONS

This section details the simulation procedure for the described work. The procedures flow chart is presented in Figure 5. The paper covers the process of data generation and introduces three detector generation methods, which will be compared. The framework for the research is

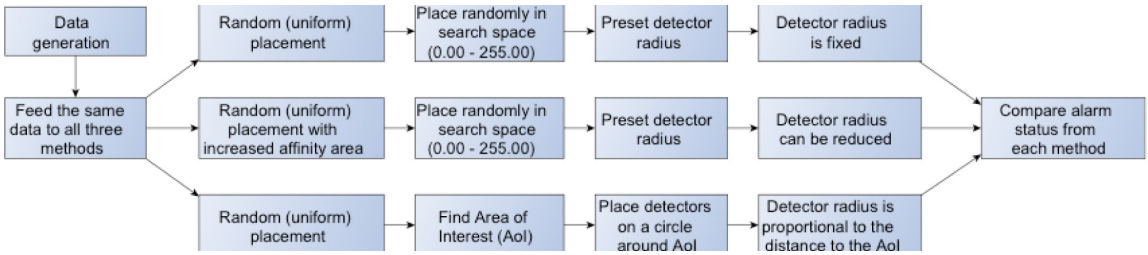
- AIS does not know the area of interest.
- Limited to 2 attributes -> 2D search space.
- 25500 possible attribute values (Data value ranges from 0.00 to 255.00).

Data Generation

To have full control over the analysis of the AIS the authors did not use any previously collected sensor data but decided to generate their own pseudo-random data set, as long-long term data is not easy to collect, for reasons described above. Furthermore, this allows the research team to know when an abnormal data pattern accurse and how it is represented in the sensor data. The key attributes for the data were:

- The normal data pattern is frequent.
- The abnormal data pattern infrequent.
- A new data point arrives every 0.2 seconds.

Figure 5. Simulation flowchart



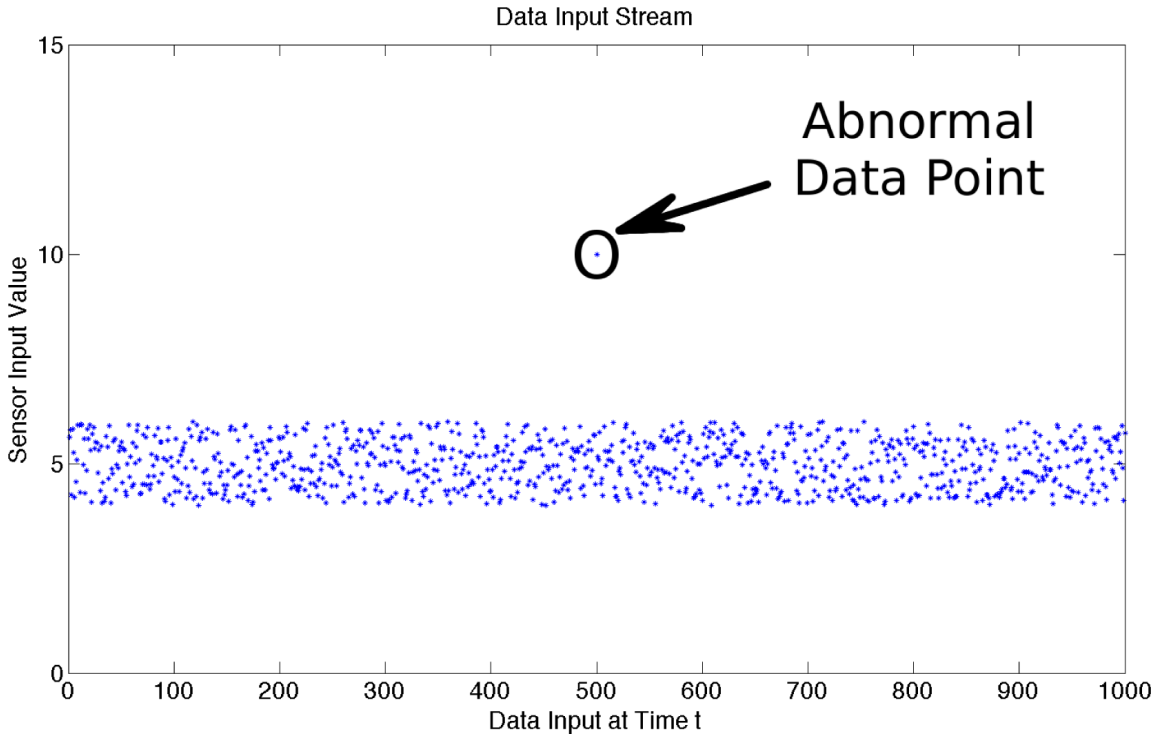
For reproducibility of the test scenario, the normal input data are float values, rounded to two decimal points, in the range of 4 to 6, uniformly distributed. For the abnormal data a pseudo-random process was used to generate every 4 days an abnormal data point, with selected value doublets of [10 10] [100 100] [170 10] [10 220] [200 200], which were introduced in the data stream. The randomisation process determines the time frame in the day when to introduce the abnormal data point. That way each abnormal day will look slightly different.

Another option for a more advanced pseudo-random process is to not include particular "abnormal" days. That way not only the time in the day would be randomised but also the day itself would be random. While the work described here does not utilize this method, future work of the authors will include this method. The values used were selected at random. As described above the AIS does not know what normal and abnormal data values are, changes to these values should not lead to changes in the simulation outcomes. Figure 6 shows an example of the data stream generated.

Random Placement

The random placement (RP) method used, places detectors, as the name suggests, at random positions in the search space using a uniform distribution. Each detector receives a doublet value to determine its centre position in the search space (in a 3D environment it would receive a triplet value). The data pair has a possible float value in the range of 0.00 to 255.00. With the simulation configuration used there would be 650 250 000 possible detector positions. Covering each position with an individual detector would have an enormous impact on the systems resources and is unnecessary as the affinity area is covering the adjacent area around each detector. Therefore, the authors arbitrarily limited the number of detectors for this method to 100. As mentioned above, the RP method used a uniform distribution to generate doublet values to populate the search space. Additionally, each detector has a fixed affinity area. This affinity area declares the monitored area around the particular detector. Input data that falls into this area will activate the detector.

Figure 6. Example of data stream



For a better visualisation Figure 7 shows the affinity area (yellow) around each detector's centre position. The graphical illustration allows for an easy detection of holes in the search space. Noticeable also, is the overlap of some detector affinity areas, which is the result of detectors being placed too close to each other. Something that is also visible is the small size of the area of interest in comparison to the rest of the search space. Changes to the size or position of the area of interest should not have any influence on the area of interest, as the AIS is an unsupervised learning method.

Variable Affinity Area

The variable affinity area (VAA) method investigated as part of this paper is similar to the first method. It is also based on randomly placed detectors. Therefore each detector will receive a doublet float value between the ranges of 0.00 to 255.00 to define the detectors centre point. Furthermore, the same uniform distribution for an equal allocation in the search space is used. The difference between the VAA and RP methods is that the former adopts the following changes in the algorithm to determine the size of the affinity area:

- Start with increased affinity area.
- Let affinity area shrink down to a threshold level.

The idea behind this method is that an increased affinity area increases the probability of a fully covered search space, while the option to shrink the affinity area should provide better coverage around the area of interest. That way, detectors that are placed too close to the area of interest are not instantaneously removed from the pool but can first decrease the monitored area around them before a threshold defines that the detector was placed to close to normal data. Figure 8 illustrates the affinity areas (yellow) around each detector with again a limited detector count of 100 at the beginning of the simulation. It can be seen that the coverage of the search space increased, while still not

Figure 7. Search space coverage using random placement

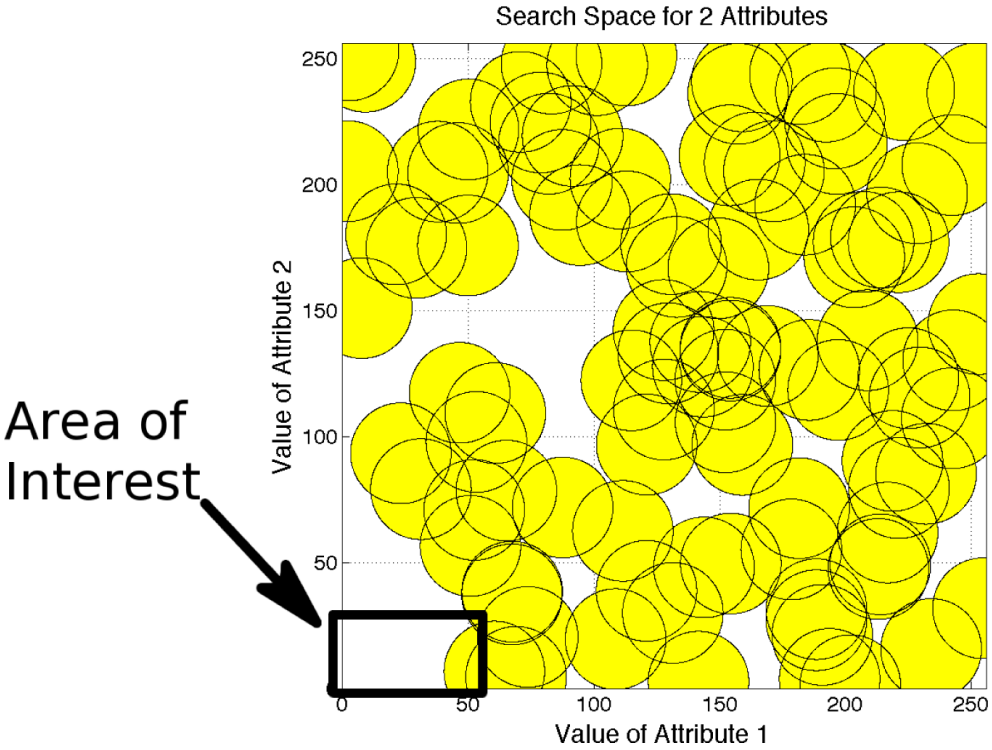
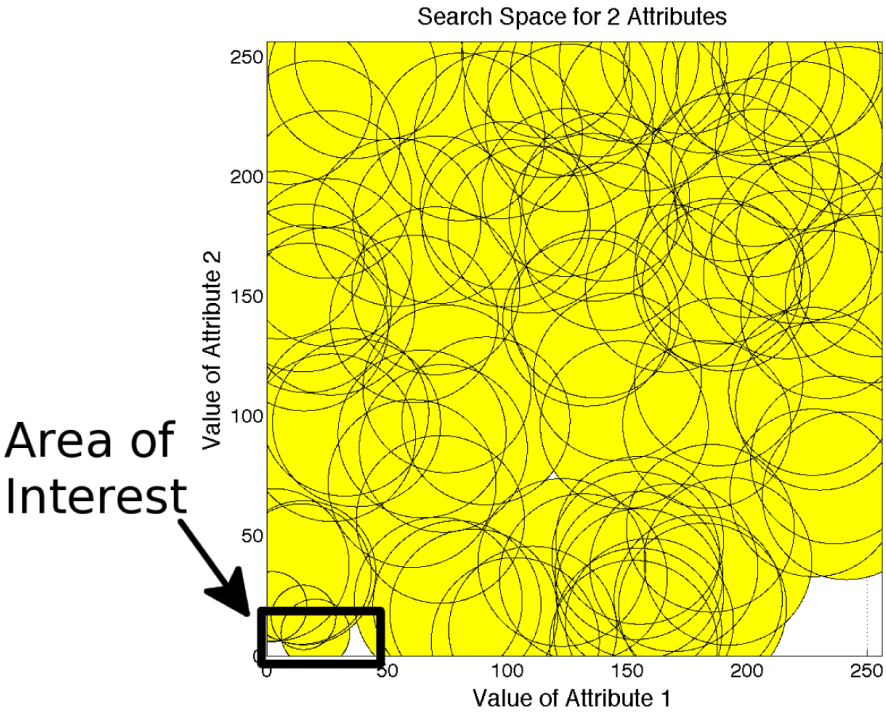


Figure 8. Search space coverage using random placement and variable affinity area



fully covered. Additionally, the overlap of the affinity areas increased as well at the same time. Therefore, if detectors are placed in a more intelligent way, it will not only reduce overlapping but would furthermore help to reduce the amount of detectors needed to cover the complete search space.

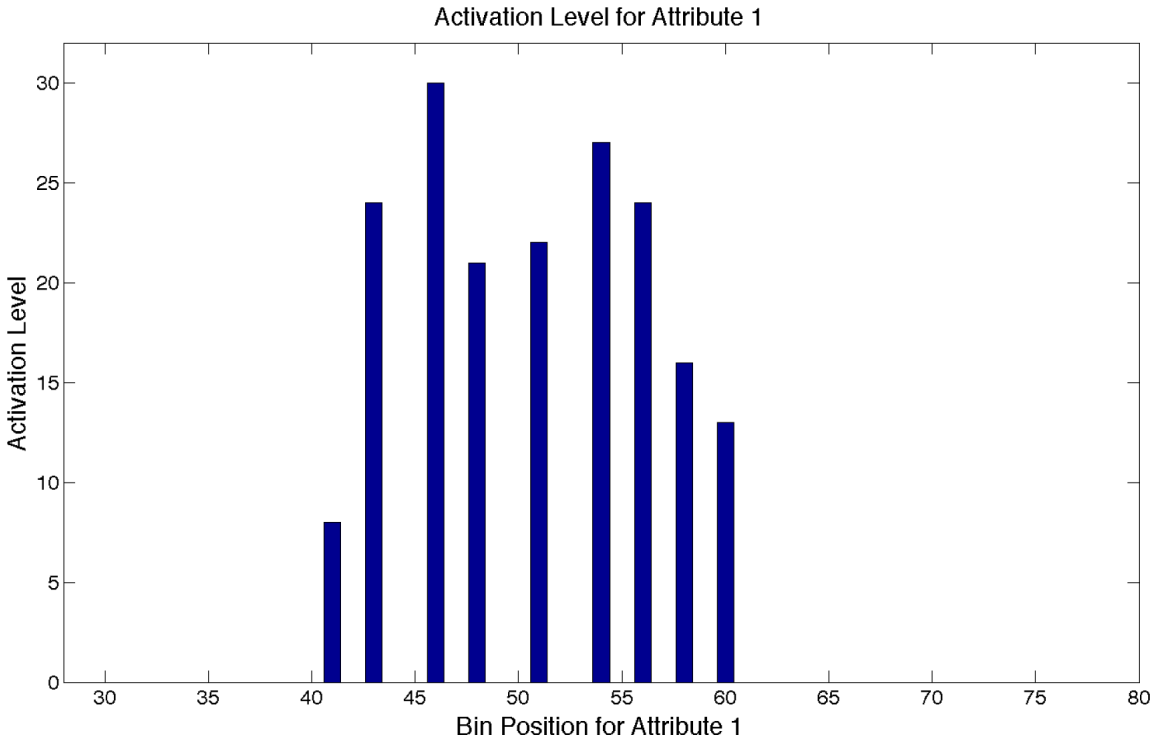
Centre Point Seeding

Because of the apparent limitations of the above methods in terms of coverage, the third solution, centre point seeding (CPS), investigated as part of this work places most detectors in the area around the normal data and only a few detectors in the rest of the search space. Furthermore, the affinity area around a detector should depend on its distance from the area of interest. Consequently detectors closer to this area should have a small affinity area, while detectors further away should cover a much wider area based on the assumption that data outside the normal area is an infrequent event.

The authors have developed a method to detect the area of interest and start placing detectors around this centre point. The two-step process starts with identifying the area of interest and then placing detectors around this area. The first simulated day is used to identify the area of interest but later research will address changes in the behavior and will therefore need to evaluate the area of interest repeatedly (such as once a day).

In the first stage the continuous input data is discretised. The possible bin values range from 0.00 to 255.00 for each attribute. A continuous background process keeps count of each individual input bin (see activation level in Figure 9). This input count is later used to detect the possible area of interest, when the activation level of the individual bin is compared to a zero activation level. In Figure 9 only one area of interest is visible but additional areas could be at other positions. Therefore the first step is to get all possible areas of interest for each attribute, in a second step these areas can

Figure 9. Activation level of bins



be analysed in more detail to determine their maximum activation level, which translates to the centre point of the activation area.

Once these areas are specified, detectors are placed around the centre points. The position of detectors is based on Equations (1) and (2).

$$R_x = X_{Detector_{CentrePoint}} + radius * \cos(\vartheta) \quad (1)$$

$$R_y = Y_{Detector_{CentrePoint}} + radius * \sin(\vartheta) \quad (2)$$

A circle with radius x is divided into *y* points on the circumference. At every iteration the radius *x* is increased whilst *y* (the number of points on the perimeter, here 20) stays constant. Therefore the distance between each of the y points is increased. The next step is to use the position of the *y* points to place detectors (see Figure 10). If the detectors were generated with a constant affinity area, this would lead to massive gaps in the search space. Therefore the radius of the affinity area is proportional to the distance of the area of interest. This results in a denser coverage around the area of interest while detectors further out can monitor a wider area. Additionally, this layout prevents the problem of massive local detector overlap present in the RP and VAA positioning methods. The CPS method distributes the overlap more equally over the whole search space. As a result less detectors are needed for the same coverage (see Figure 11 showing only the detector positions and Figure 12 showing the detector affinity areas around the area of interest).

RESULTS AND DISCUSSION

After introducing these three described methods in the simulation condition section above, the authors validated their optimization in detector seeding using simulation data. All three methods got the same input stream with the same abnormal data point every 4 days. The performance measures used were:

Figure 10. Calculation of detector positions

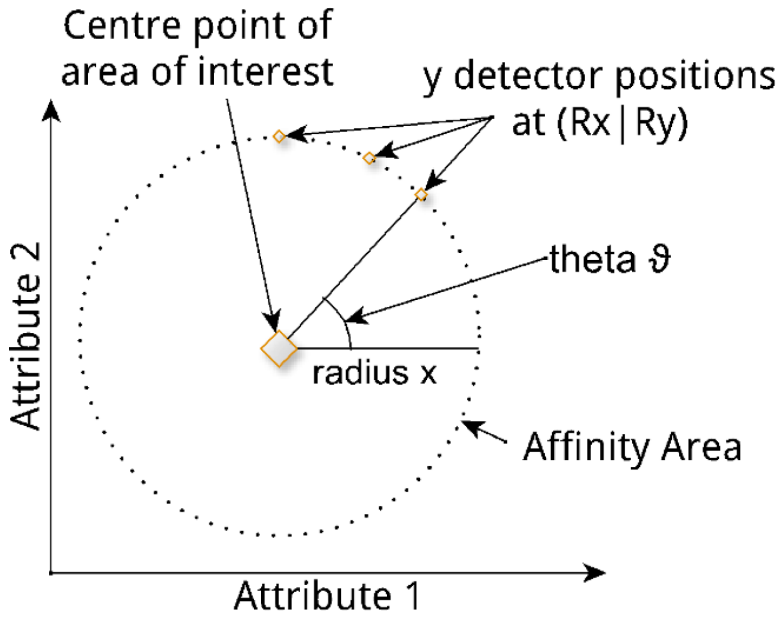


Figure 11. Detector placement in the search space

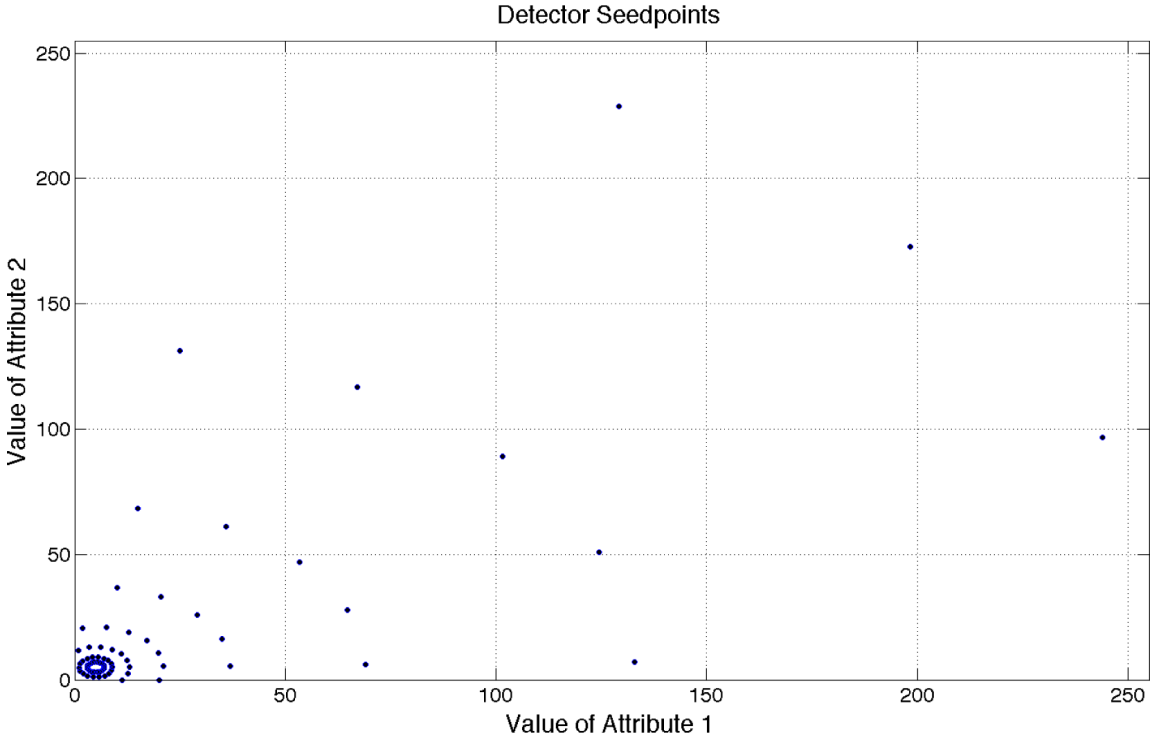
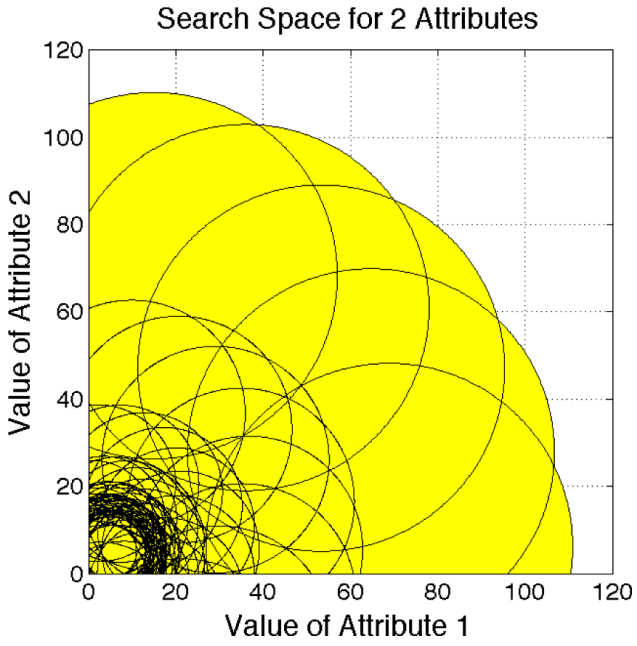


Figure 12. Detector placement in the search space



- Correct classification of abnormal data point.
- Missing abnormal data point.
- Incorrect classification of normal data point.

The authors ran 20 different simulations, each lasting 20 simulated days with 5 abnormal data points to detect per runtime. The position of the abnormal data was preselected. Table 1 shows the results of the simulation.

The results show that the RP and VAA methods achieve nearly the same results, while the VAA method seems to have resulted in a slight improvement. The reason for missed anomalies is the described placement of the detectors. During individual runtimes of the simulation, the uniform distribution leads to

Table 1. Simulation results

	RP	VAA	CPS
Total	100	100	100
Correct	30	34	100
Missing	70	66	0
Incorrect	0	0	0

different positions of the detectors and therefore to changes in the search space coverage, which results in undetected abnormalities.

On the other side, the placement of detectors using the CPS method achieved a 100% detection rate. What the results also highlight is that none of the methods alarm incorrectly. Furthermore CPS reduce the occurrence of false negatives, this is important in terms of the user acceptance rate for such a system, and would lead to longer wear time (by the user), as the system will be perceived as being reliable.

The foundation for good anomaly detection is a fully covered search space and this becomes one of the key aspects for an Artificial Immune System in the area of Ambient Assisted Living. Therefore, placing detectors using a more intelligent approach around the area of interest, the system achieves an increased confidence of detecting abnormalities. Also the self-learning feature of AIS proves to be promising, which could lead to an elderly monitoring system that automatically adapts to its owner without costly data collection and annotation beforehand.

CONCLUSION AND FUTURE WORK

This is paper has presented a possible approach for anomaly detection and long term monitoring in Ambient Assisted Living (AAL) applications using Artificial Immune Systems (AIS). A review of the literature has highlighted the successes achieved through the use of AIS in intrusion detection in computer networks, fault detection of electronic machines, and unsupervised classification of different data sets. The authors have made the case for how AIS can be applied in ambient assisted living scenarios to address the problem of the long term monitoring of elderly people. An improved method for the optimisation of detector generation for AIS, which uses a novel intelligent seeding technique, has been presented. The new seeding technique (CPS) has been compared with two other methods (RP and VNN) and the simulation results have been presented and discussed.

The new technique resulted in an improved coverage of the area of the search space through the intelligent management of detector affinity area overlap. The technique has also resulted in a reduced detector count, which helps lower the load on available resources, thus improving both runtime and overall battery life of what will ultimately be a wearable, non-invasive monitoring system. Overall the use of AIS in area of AAL has been shown, through the results of the simulation work presented, to be a promising approach worth pursuing.

The authors are currently investigating whether the CPS method can be used for multi dimensional search spaces and ways in which obsolete detector can be used to improve the seeding process by generating better immature detectors.

REFERENCES

Bersch, S., Azzi, D., & Khusainov, R. (2011, Aug 3). Fall Detection using Biologically Inspired Monitoring. eprints.port.ac.uk, 1-4.

Bersch, S., Chislett, C., Azzi, D., & Khusainov, R. (2011, Jan 1). Activity detection using frequency analysis and off-the-shelf devices: fall detection from accelerometer data. eprints.port.ac.uk. Retrieved from http://eprints.port.ac.uk/2456/

Bersini, H., & Carneiro, J. (2006). *Artificial Immune Systems: 5th International Conference, ICARIS 2006, Oeiras, Portugal, September 4-6, 2006, Proceedings (Lecture Notes in Computer Science)*.

Boylan, G. (2011, Jan 1). EEG monitoring in the neonatal intensive care unit: A critical juncture. *Clinical neurophysiology: official journal of the ...*. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/21530388

Cai, Y., Chow, M.-Y., Lu, W., & Li, L. (2010). Evaluation of distribution fault diagnosis algorithms using ROC curves. *Power and Energy Society General Meeting, 2010 IEEE*, 1 - 6. Retrieved from http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=5588154&queryText%3D%28%28artificial+immune+system%29%29%26openedRefinements%3D*%26sortType%3Ddesc_Publication+Year%26matchBoolean%3Dtrue%26rowsPerPage%3D50%26searchField%3DSearch+All

Commission, E., Economic, D.-G., & Affairs, F. (2009). Ageing Report 2009. Retrieved from http://bookshop.europa.eu/is-bin/INTERSHOP.enfinity/WFS/EU-Bookshop-Site/en_GB/-/EUR/ViewPublication-Start?PublicationKey=KCAR09002

Elbert, D., Storf, H., & Eisenbarth, M. (2011, Jan 1). An approach for detecting deviations in daily routine for long-term behavior analysis. *Pervasive-Health*. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6038843

Gianini, G., Anisetti, M., Azzini, A., Bellandi, V., Damiani, E., & Marrara, S. (2009). An artificial immune system approach to anomaly detection in multimedia ambient intelligence. *Digital Ecosystems and Technologies, 2009. DEST '09. 3rd IEEE International Conference on*, 502 - 506. Retrieved from http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=5276702&queryText%3D%28%28Artificial+Immune+System%29+AND+%28Ambient%29%29%26openedRefinements%3D*%26sortType%3Ddesc_Publication+Year%26matchBoolean%3Dtrue%26rowsPerPage%3D50%26searchField%3DSearch+All

Golovko, V., Komar, M., & Sachenko, A. (2010). Principles of neural network artificial immune system design to detect attacks on computers. *Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), 2010 International Conference on*, 237 - 237. Retrieved from http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=5446089&queryText%3D%28%28artificial+immune+system%29%29%26openedRefinements%3D*%26sortType%3Ddesc_Publication+Year%26matchBoolean%3Dtrue%26rowsPerPage%3D50%26searchField%3DSearch+All

Hofmeyr, S., & Forrest, S. (2000). Architecture for an artificial immune system. *Evolutionary Computation*, *8*(4), 443–473. doi:10.1162/106365600568257 PMID:11130924

Li Zhi-tang. Li Yao, & Wang Li. (2005). A novel fuzzy anomaly detection algorithm based on artificial immune system. *High-Performance Computing in Asia-Pacific Region, 2005. Proceedings. Eighth International Conference on*, 5 pp. - 486. Retrieved from http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=1592309&queryText%3D%28%28Artificial+Immune+System%29+AND+%28living%29%29%26openedRefinements%3D*%26sortType%3Ddesc_Publication+Year%26matchBoolean%3Dtrue%26rowsPerPage%3D50%26searchField%3DSearch+All

Monekosso, D., & Remagnino, P. (2010). Behavior Analysis for Assisted Living. *Automation Science and Engineering, IEEE Transactions on, PP*(99), 1 - 1. Retrieved from http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=5475363&queryText%3D%28%28sensor+processing%29+AND+%28assisted+living%29%29%26openedRefinements%3D*%26sortType%3Ddesc_Publication+Year%26matchBoolean%3Dtrue%26rowsPerPage%3D50%26searchField%3DSearch+All

Polat, K., & Gunes, S. (2006, Jan 1). Diagnosis of heart disease using artificial immune recognition system and fuzzy weighted pre-processing. *Pattern recognition*. Retrieved from http://www.sciencedirect.com/science/article/pii/S0031320306002445

Sim, K., Yap, G.-E., Phua, C., Biswas, J., Phyo Wai, A., Tolstikov, A., et al. (2010). Improving the accuracy of erroneous-plan recognition system for Activities of Daily Living. *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*, 28 - 35. Retrieved from http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=5556555&queryText%3D%28%28learning%29+AND+%28assisted+living%29%29%26openedRefinements%3D*%26sortType%3Ddesc_Publication+Year%26matchBoolean%3Dtrue%26rowsPerPage%3D50%26searchField%3DSearch+All

Virone. (2008). Monitoring activity patterns and trends of older adults. *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, 2071 - 2074. Retrieved from http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=4649600&queryText%3D%28%28Authors%3AG+Virone%29%29%26openedRefinements%3D*%26sortType%3Ddesc_Publication+Year%26matchBoolean%3Dtrue%26rowsPerPage%3D50%26searchField%3DSearch+All

Sebastian Bersch received a Diplom Ingineur (FH) degree in electronic engineering from the University of Applied Science Münster, Germany in 2009 and an M.Eng, degree in electronic engineering from the University of Portsmouth, UK in 2010. Currently, he is a PhD Student in the School of Engineering at the University of Portsmouth with his research topic in biological inspired AI methods for assisted living.

Djamel Azzi received a Ingénieur D'État degree in electronics from the Université de Constantine (Algeria) in 1988 and a M.Sc. degree in control technology (1991) and a Ph.D. degree in control engineering (1993) from the University of Manchester Institute of Science and Technology (UMIST) Manchester U.K. He is currently a Principal Lecturer in the School of Engineering at the University of Portsmouth U.K. His research interests include all aspects of AI but in particular its application to robotics, wireless sensor networks and assisted living.

Rinat Khusainov received a Diploma degree in Computer Engineering from Chelyabinsk State Technical University, Russia and M.Sc and Ph.D degrees in Computer Science from the University College Dublin, Ireland. Currently, he is a Senior Lecturer in the School of Engineering at the University of Portsmouth, UK. His research interests include Artificial Intelligence, Machine Learning, Distributed and Multi-Agent Systems, Information Retrieval, and applications of these methods to Assisted Living.

Ifeyinwa E. Achumba (formerly Chika) holds the B.Sc degree in Computer Science, the M.Sc, degree in Computer Engineering, and the PhD degree in Electronic and Computer Engineering. Currently, Ifeyinwa Achumba is a research associate in the Electronic and Computer Engineering Department of the University of Portsmouth (UoP), UK, in a postdoctoral capacity.

FORM UPR16

Research Ethics Review Checklist

Please complete and return the form to Research Section, Quality Management Division, Academic Registry, University House, with your thesis, prior to examination

Postgraduate Research Student (PGRS) Information			Student ID:	407189
Candidate Name:	**Sebastian Dominik Bersch**			
Department:	**ENG**	**First Supervisor:**	**Djamel Azzi**	
Start Date: (or progression date for Prof Doc students)	**1. May 2010**			

Study Mode and Route:	Part-time ❑	MPhil ❑	Integrated Doctorate (NewRoute) ❑
	Full-time ⊟	MD ❑	Prof Doc (PD) ❑
		PhD ⊟	

Title of Thesis:	Artificial Immune System For The Detection of Abnormal Activity In Ambient Assisted Living
Thesis Word Count: (excluding ancillary data)	47000

If you are unsure about any of the following, please contact the local representative on your Faculty Ethics Committee for advice. Please note that it is your responsibility to follow the University's Ethics Policy and any relevant University, academic or professional guidelines in the conduct of your study

Although the Ethics Committee may have given your study a favourable opinion, the final responsibility for the ethical conduct of this work lies with the researcher(s).

UKRIO Finished Research Checklist:
(If you would like to know more about the checklist, please see your Faculty or Departmental Ethics Committee rep or see the online version of the full checklist at: http://www.ukrio.org/what-we-do/code-of-practice-for-research/)

a) Have all of your research and findings been reported accurately, honestly and within a reasonable time frame?	**YES**
b) Have all contributions to knowledge been acknowledged?	**YES**
c) Have you complied with all agreements relating to intellectual property, publication and authorship?	**YES**
d) Has your research data been retained in a secure and accessible form and will it remain so for the required duration?	**YES**
e) Does your research comply with all legal, ethical, and contractual requirements?	**YES***

*Delete as appropriate

Candidate Statement:	
I have considered the ethical dimensions of the above named research project, and have successfully obtained the necessary ethical approval(s)	
Ethical review number(s) from Faculty Ethics Committee (or from NRES/SCREC):	N/A
Signed: ***(Student)***	**Date:**
If you have *not* submitted your work for ethical review, and/or you have answered 'No' to one or more of questions a) to e), please explain why this is so:	
The work of the PhD research project was deemed to not require an ethical review.	
Signed: ***(Student)*** Sebastian Bersch	**Date: 8. September 2014**